

WL-TR-94-3114

A THREE-DIMENSIONAL INVISCID FLOW  
SOLVER IN CHIMERA FLOW SIMULATION



Y. WANG

UNIVERSITY OF CALIFORNIA, DAVIS  
DEPT OF MECHANICAL, AERONAUTICAL AND  
MATERIALS ENGINEERING  
DAVIS CA 95616-5294

FEBRUARY 1994

FINAL REPORT FOR 08/15/90-02/15/94

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



19950518 038

FLIGHT DYNAMICS DIRECTORATE  
WRIGHT LABORATORY  
AIR FORCE MATERIEL COMMAND  
WRIGHT PATTERSON AFB OH 45433-7562

DTIC QUALITY INSPECTED 5

# DISCLAIMER NOTICE

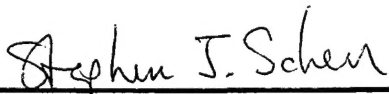


THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

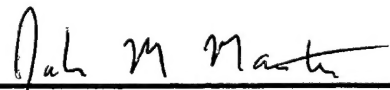
## NOTICE

WHEN GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA ARE USED FOR ANY PURPOSE OTHER THAN IN CONNECTION WITH A DEFINITELY GOVERNMENT-RELATED PROCUREMENT, THE UNITED STATES GOVERNMENT INCURS NO RESPONSIBILITY OR ANY OBLIGATION WHATSOEVER. THE FACT THAT THE GOVERNMENT MAY HAVE FORMULATED OR IN ANY WAY SUPPLIED THE SAID DRAWINGS, SPECIFICATIONS, OR OTHER DATA, IS NOT TO BE REGARDED BY IMPLICATION, OR OTHERWISE IN ANY MANNER CONSTRUED, AS LICENSING THE HOLDER, OR ANY OTHER PERSON OR CORPORATION; OR AS CONVEYING ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY IN ANY WAY BE RELATED THERETO.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.



STEPHEN J. SCHERR  
Project Engineer  
CFD Research Section



JOSEPH M. MANTER  
Chief  
CFD Research Branch



DENNIS SEDLOCK  
Chief  
Aeromechanics Division

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION PLEASE NOTIFY WL/FIMC, WRIGHT-PATTERSON AFB, OH 45433-7562 TO HELP MAINTAIN A CURRENT MAILING LIST.

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIC DOCUMENT.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE FEB 1994	3. REPORT TYPE AND DATES COVERED FINAL 08/15/90--02/15/94		
4. TITLE AND SUBTITLE A THREE-DIMENSIONAL INVISCID FLOW SOLVER IN CHIMERA FLOW SIMULATION		5. FUNDING NUMBERS C F33615-90-C-3004 PE 61102 PR 2307 TA N6 WU 14		
6. AUTHOR(S) Y. WANG				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSITY OF CALIFORNIA, DAVIS DEPT OF MECHANICAL, AERONAUTICAL AND MATERIALS ENGINEERING DAVIS CA 95616-5294		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) FLIGHT DYNAMICS DIRECTORATE WRIGHT LABORATORY AIR FORCE MATERIEL COMMAND WRIGHT PATTERSON AFB OH 45433-7562		10. SPONSORING/MONITORING AGENCY REPORT NUMBER WL-TR-94-3114		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) APPLICATION OF COMPUTATIONAL FLUID DYNAMICS (CFD) TO COMPLEX THREE-DIMENSIONAL CONFIGURATIONS IS CONSTRAINED BY GENERATION OF THE COMPUTATIONAL GRID, ACCURACY LIMITATIONS IMPOSED BY THE DISCRETIZATION OF THE GOVERNING EQUATIONS, AND REQUIREMENTS FOR COMPUTATIONAL EFFICIENCY. THE CHIMERA GRID TECHNIQUE SEEKS TO TAKE ADVANTAGE OF THE COMPUTATIONAL EFFICIENCY AND NUMERICAL ACCURACY DEVELOPED FOR STRUCTURED-GRID CFD TECHNIQUES AND TO TAKE ADVANTAGE OF THE GEOMETRIC FLEXIBILITY OFFERED BY OVERSET GRID MECHANISMS. THE CHIMERA METHOD CAN PROVIDE THE FLEXIBILITY TO EMPLOY BOUNDARY- CONFORMAL GRIDS ON AERODYNAMIC COMPONENTS, REFINE THE MESH SELECTIVELY IN REGIONS OF INTEREST, AND ALLOW DIFFERENT FLOW MODELS TO BE SOLVED IN DIFFERENT ZONES OF THE GRID. THE PRIMARY OBJECTIVE OF THIS RESEARCH IS TO DEVELOP THE METHODOLOGY FOR USING THE CHIMERA TECHNIQUE ON GENERAL THREE-DIMENSIONAL CONFIGURATION PROBLEMS. THIS WORK FOCUSES PRIMARILY ON INVISCID ANALYSIS, BOTH AS PROOF-OF-CONCEPT AND DEVELOPMENT FOR ZONAL CFD TECHNIQUES.				
14. SUBJECT TERMS CFD, CHIMERA GRID, ZONAL CFD		15. NUMBER OF PAGES 59		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

# TABLE OF CONTENTS

Page

1.0	INTRODUCTION	
1.1	Objective.....	1
1.2	Literature Review.....	5
2.0	CHIMERA APPROACH AND GRID TOPOLOGY	
2.1	Introduction.....	8
2.2	Chimera Method.....	12
2.2.1	Hole Creation.....	13
2.2.2	Searching for Interpolation Elements.....	15
2.2.3	Interpolation Points.....	19
2.2.4	Interpolation Methods.....	20
2.2.5	Date Structure.....	21
2.3	Grid Generation and Composit Grid.....	22
2.4	Solution Procedure.....	24
3.0	FLOW SOLVER	
3.1	Subsonic Nonconservative Formulation.....	26
3.2	General Coordinates.....	28
3.3	Numerical Algorithm.....	29
3.4	Bernoulli Equation.....	31
3.5	Boundary Conditions.....	32

iii

Accession For	
NTIS	<input checked="" type="checkbox"/> CRA&I
DTIC	<input type="checkbox"/> TAB
Unannounced <input type="checkbox"/>	
Justification .....	
By .....	
Distribution / .....	
Availability Codes	
Dist	Avail and / or Special
A-1	

4.0	VERIFICATION AND APPLICATION	
4.1	Potential Solution Around a Sphere.....	34
4.2	Solutions for Flow Around a Sphere in Single Grid.....	35
4.3	Flow Solutions with Multioverset Grids.....	37
4.4	Flow Solutions with Multibody Configurations.....	44
5.0	CONCLUDING REMARKS	50
6.0	REFERENCES	51

Figure	Page
1. Simple geometric configuration of orbiter.	2
2. Complex space shuttle.	2
3. Grid on a concave surface.	11
4. Grid on a convex surface.	11
5. Normal vector to the initial hole boundary.	14
6. Construction of fringe points.	14
7. A sphere surface covered by a single piece of grid.	16
8. The whole sphere surface covered by two pieces of grids	17
9. The whole sphere surface covered by four pieces of grids	18
10. Potential solution on a sphere surface.	38
11. Numerical solution on a sphere surface.	39
12. Potential solution on a symmetric section of sphere.	40
13. Numerical solution on a sphere symmetric section.	41
14. Mach number contour on sphere surface for four grids.	42
15. Numerical solution on a sphere surface.	43
16. One of two grids with a hole in it for two sphere.	45
17. Velocity vector in a cross section of two spheres.	46
18. Pressure contour in a cross section of two spheres.	47
19. Pressure contour in a cross section of two spheres.	48
20. Pressure contour in a cross section of two ellipsoids.	49

## 1.0 INTRODUCTION

### 1.1 Objective

Because Computational Fluid Dynamics (CFD) can treat the nonlinear phenomenon in high-Reynolds-number flow simulation with the aid of high speed digital computer, CFD are becoming in the matured stage for both physical research problems and practical engineering problems, it is becoming an integral part of the study of fluid mechanics and the aerodynamics design process [39]. Research topics on CFD are generally directed toward solving problems for understanding flow physics, treating realistic configurations, maintaining good computational efficiency and stability, postprocessing of the computer results, and maintaining user friendliness. Many researchers have been searching for methods to simulate flows around very complex geometries, such as the integrated space shuttle configuration [Ref. 21], and full aircraft configurations [Ref. 25], which depends on the use of different technologies developed for CFD and other related fields, such as grid generation, flow solver, and flow visualization.

Currently, with limited computation resources, the methods of treating realistic configurations and increasing accuracy of the numerical simulation are one of the most important tasks in the computational fluid dynamics research and practical engineering application. For a single body shape, as shown in Figure 1 for space shuttle orbiter, the use of a single body-conforming curvilinear mesh leads to the most efficient solution procedure. However, for more complex shapes, especially for complex three-dimensional configuration, as shown in Figure 2, for a simplified space shuttle configuration, it is quite difficult to generate a single body-conforming well-clustered curvilinear grid that is not overly skewed and has smooth variation. On the other hand, the accuracy enhancement is another obstacle in the computational fluid dynamics. No matter what discretization methods would be used, finite difference or finite volume, the solution of the partial

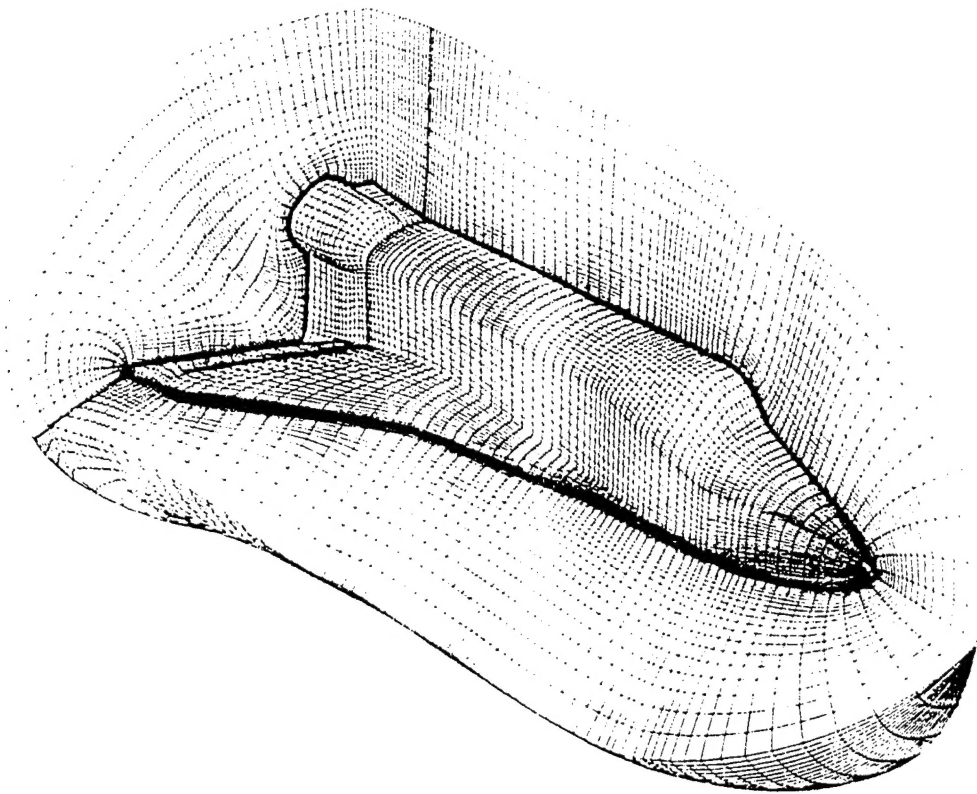


Figure 1. Simple geometric configuration of orbiter

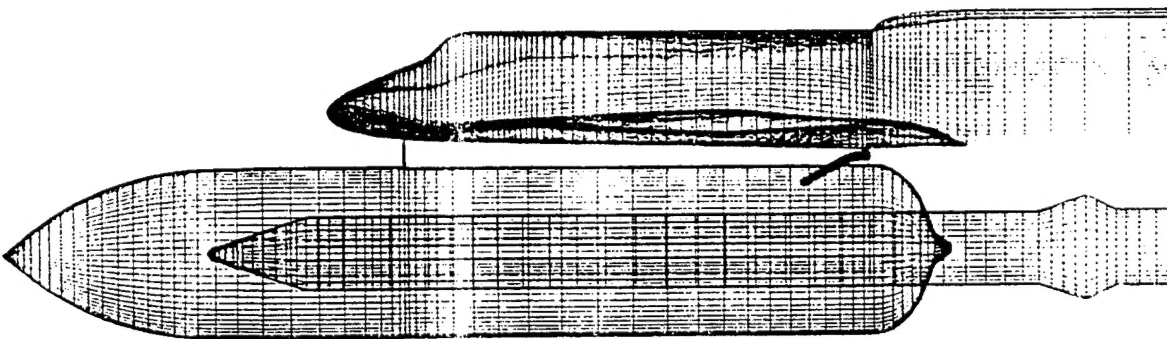


Figure 2. Complex space shuttle

differential equations of fluid motion requires that the computational domain and dependent variables be represented on a network of discrete points. The distribution of these points is influenced by the choice of the coordinate system, the order of the numerical approximation, and the location of strong geometric and flow field gradients. As long as the discretization method is used, the solutions of the differential equations become more accurate as the number of the grid points are increased. But because of the limitation of computer resources, it is necessary to improve the grid resolution without increasing the number of grid points.

Many different methods have been developed in grid generation and discretization to improve the grid resolution. Grid adaptation is one of the solutions. Local grid resolution can be improved by adopting solution-adaptive grid technique without increasing the total number of grid points. Another method is local grid enrichment. In addition to the original grid, locally defined finer grid is created and overlapped or overlaid on the region of interest of the original grid.

For geometry imposed complex flow fields, composite structured grid schemes and unstructured grid schemes are two major approaches for nonlinear simulation. For the most part, the unstructured grid approaches have come from the finite element community, although finite difference and finite volume methods can be written for an unstructured grid. Both approaches require a more complex data-handling program than that required by a simple single-structured grid, and both approaches have their strengths and weaknesses. It seems that unstructured grid methods are generally considered to be more versatile and easier to adapt to complex geometry, but it has difficulty in treating viscous flow problems, while composite structured grid methods are generally considered to use more efficient numerical algorithms, require less computer memory, and can handle viscous flow problems without difficulty. The best approach is still not resolved, but hybrid schemes, which use is structured grid in most areas and only unstructured grid in limited regions, have already appeared [Refs. 40 through 42], it incorporates the best features of both

schemes, and will certainly grow in popularity and ultimately prove to be the best for treating complex configurations.

Composite grids use more than one grid to mesh an overall configuration with each individual grid of the system patched or overset together. Patched grids are individual meshes that are joined together at some common interface plane, but it is still relatively difficult in grid generation because various interfaces have to be defined and grids have to be generated with both inner and outer defined boundary surfaces. Overset grids do not require a common interfaces, but rather, a simply superimposed or partially superimposed mesh is needed to cover the region of interest, and to provide the means of matching the solutions across boundary interface. Without any special way of grid connection or common interface, the grid generation is greatly simplified since the hyperbolic grid generation technique may be used.

The composite overset or Chimera method can provide the flexibility to employ boundary conforming grids on component parts of the geometry, define the mesh selectively in the regions of interest, and permit different flow models to be solved on each grid. In this discretization process, each grid in the system is ordered and is thus suitable for efficient finite difference solutions on vector computer and any single-grid code. The primary objective of present study is to develop a composite overset or Chimera code which can handle general 3-D configuration problems.

For a complex geometric configuration problem, to generate composite overset grids, and construct required data structure for Chimera scheme is not an easy job, and it is more costly to run a three-dimensional Chimera scheme for Navier-Stokes equations. Therefore, before running the Navier-Stokes solver, it is very important to check Chimera data structure to make sure that all the Chimera overset grids connections are working correctly. Therefore, besides developing a Chimera connecting code, the other purpose of present work is to develop a simple numerical procedure in three dimension, which can easily fit in Chimera scheme and give an initial inviscid flow solution. These inviscid

results from the simple flow solver can be used to check the Chimera scheme, and to be used as the initial results of Navier-Stokes solver.

## 1.2 Literature Review

With the aid of supercomputer, CFD researchers start to calculate complex configurations, e.g., Meakin, et al. [Refs. 17, 18, and 19], used overset grid method to simulate transient flow field by solving Navier-Stokes equations for integrated space shuttle vehicle configuration during ascent and SRB separation. Buning, et al. [Refs. 21, 22 and 23], used an overset grid method to solve the thin-layer Navier-Stokes equations for the integrated space vehicle launching configuration. Flores, et al. [Ref. 56], solved the Navier-Stokes equations for a transonic flow by using a patched grid method. Obayashi [Ref. 57] solved Navier-Stokes equations for a wing-fuselage-tail geometric configuration by using a single grid system, and Jameson, et al. [Ref. 58], solved Euler equations for Boeing 747-200 using an unstructured grid.

Of these calculations of complex geometries, the grid system can be roughly divided into the following types:

- single grid
- composite grids, which can be divided into:
  - patched grid
  - overset grid
  - adaptive grid

For a single grid, it is simple to solve control equations but it is difficult for grid generation for a complex geometric configurations, and it may contain overly skewed meshes, which could give an inaccurate solution. On the other hand, using a single grid could not refine the grid selectively, therefore, it may require more grid points than a

comparable multiple grid approach. Comparing with a single grid, a multiple grid method is flexible, but it also demands a significant amount of computer resources and different techniques, which may involve computational domain-decomposition, connection or communication between each grid and composite grids data management.

Domain-decomposition techniques subdivide the computational domain into simpler subdomains which admit a more easily constructed mesh. Several strategies have been explored to subdivide the domain and establish communications among the subdomains. One group of approaches, the grid-patching or zonal methods, uses common or shared boundaries and another uses embedded or overset grids to subdivide the domain. The work of Rubbert and Lee [Ref. 56] is typical of the methods which construct a global mesh from subdomains which share common subdomains. They generate a global mesh by solving grid-generation equations on all subdomains simultaneously and by requiring that the grid lines be continuous across subdomain boundaries. A difficulty with this approach is that irregularities which occur in corners and along boundaries impose constraints on the algorithm used to solve the flow equations. Lasinski [Ref. 43] takes an alternate approach and solve for the flow field on each subdomain separately with communication among the grids established by the transfer of boundary data. In their approach, the patches overlap one point with common points on the boundary to obviate interpolation for boundary data. Hessenius and Pulliam [Ref. 44] have modified the approach. Rai [Ref. 45] further generalized the method to admit independent grids in each subdomain. Communication across grid boundaries is accomplished by means of special difference formula at the boundaries which maintain conservation properties across the subdomains. Similar methods have been developed by Miki and Takagi [Ref. 46]. Holst, et al. [Ref. 47] have applied the technique to large 3-D grids.

The grid-embedding or oversetting techniques do not require common boundaries between subdomains, but rather, a common or overlap region is required to provide the means of matching the solutions across boundary interfaces. The usual procedure uses interpolations of embedded boundaries to provide the necessary communication among the grids. There are several implementations of the method. Steger, et al. [Ref. 33] and Benek, et al. [Ref. 4], developed a "Chimera" scheme in three dimensions for the solution of both a linearized flow model or Euler equations. Dougherty [Ref. 38] has extended the grid-embedding technique to time-dependent motions. Atta [ref. 48] applied this method to solve the full-potential equation in two and three dimensions using a separate implicit solution algorithm for each grid. Fuchs [Ref. 49] employed this approach to internal flows, and Rai [Ref. 50] used a combination of patched and overset grids to solve rotor-stator interaction problems.

The second technique, grid-adapting methods, forces the mesh to evolve with the solution of the flow equations. the advantage of this method is that the initial mesh does not need to anticipate accurately all regions of large flow gradients, thus it can make the most efficient use of available mesh points, and reduce the grid-generation effort by automatically clustering grid points to regions of high gradients. Gnoffo [Ref. 51] models the mesh as a network of springs whose constants are determined from the flow gradients. Nakahaski, et al. [Ref. 52], extend these methods to both steady and unsteady flow problems. Ghia, et al. [Ref. 53], used both the grid-evolution equation and the flow equation by requiring that the coefficient of the convective term in the flow model be minimized. Brackbill and Saltzman [Ref. 54] used variational methods to produce grid-evolution equations. Berger [Ref. 56] developed a dynamic grid refinement technique by successively embedding finer grids to resolve flow gradients.

## 2.0 CHIMERA APPROACH AND GRID TOPOLOGY

### 2.1 Introduction

For complex geometries, grid generating itself is not an easy task. Though it is possible to generate a single grid for a complex geometry, the resultant grid is most often overly skewed in certain directions or regions, or doesn't have the needed clustering to resolve the flow field in regions of rapid change. One good method to overcome this difficulty is to divide the complex shape into several simple ones and generate the grid about these simple shapes, then overset them one on the top of the others. With using inter-grid communication in the flow solver, the overset grid can be used to solve complex geometric problems.

The overset grid approach used in this research was first devised by Steger, and his colleague [Refs. 2 through 5, 10, 16, 21, 22, and 33, etc.], and given the name, the Chimera approach, after the Greek legendary creature that was compounded of incompatible parts which signifies that the Chimera approach can take incompatible grids (i.e., no common boundaries between different grids) and 'glue' them together to be solved by the flow solver. Because the Chimera approach does not require common boundaries between component meshes, complex grid generating can be replaced by a combination of a series of simple ones. Therefore, it provides the flexibility to employ boundary-conforming grids on component parts of the geometry, to change grids of some component without changing other component grids. Furthermore, it is also easy to refine the mesh selectively in the regions of interest without regenerating the whole grid for the complete configuration. Another feature of the Chimera approach is that the flow solver simulation is done in sequence for component grids, which offers a saving in memory usage for solving flows around a complex geometry since it only requires memory enough to handle

the largest component grid, and it is also possible to use different computers to update the results on each grid if a multitasking computer processor is available.

In this discretization process, each individual grid in the system is well ordered and thus suitable for efficient finite difference solution using highly vectorized computers and any available single-grid code. It is also possible to use different models or schemes with different time steps, etc., for different components, which opens many possibilities for enhancing the rate of convergence.

The Chimera method has been demonstrated for a wide variety of aerodynamic applications including, among many others, simulations about the integrated space shuttle vehicle for various ascent conditions, [Refs. 21 through 24], transonic computations about the entire F-15 aircraft with pylons and stores [Ref. 12], simulation of the viscous flow about the F/A-18 at high angles of attack [Ref. 27], and numerical studies of a delta wing platform with multiple jets in ground effects [Ref. 7]. In addition, overset grid techniques represent a powerful tool for analyzing problems involving relative motion between vehicle components. Such applications have been carried out time-accurately in three dimensions for the separation sequence of the space shuttle solid rocket boosters [Ref. 16,17 and 19], and aircraft store separation sequences [Ref. 10, 11 and 19]. The Chimera method has also been successfully applied to many nonaerodynamic problems, such as applications in biomedical fluid mechanics [Ref. 15], environmental flow simulations [Ref. 18], and the high speed train project in Japan [Ref. 13].

Although there are many advantages to the overset mesh method, there are also disadvantages which may influence the method widely use. Because interpolation points and blanked points must be located and labeled for special treatment, a massive data base is needed. These schemes are more complex relative to using a single grid scheme and data interpolation also can cause inaccuracies such as local loss of numerical conservation of fluxes. For unsteady problems with relative motion between grids, the interpolation data

and index need to be calculated at each time step, therefore, it would be more expensive than a single grid scheme.

Although there are many applications of the Chimera method, the successful code is still very limited, especially for viscous flow problems. A very good code PEGSUS (versions 2.0, 3.0 and 4.0), which was developed by CALSPAN of AEDC, has been widely used by Chimera group researchers, and has many successful applications [Refs. 21 through 23], but it still has some weaknesses needed to overcome, for example, as shown in Figure 3, the space above a concave body surface is covered by two overset grids. The values on the boundary of the solid line grid are interpolated from the solution of the dashed line grid, and the values on the boundary of the dashed grid are interpolated from the solution of the solid line grid. For PEGSUS code, the points below the surface grid are interpolated from the solution of solid line grid. For PEGSUS code, the points below the surface grid are considered inside the surface, so these points are blanked out. For viscous flow problems, the grids are very fine near the surface, thus, up to 10 points below the surface grid could be blanked out. No information can be obtained for these points. The other limitation is for high Reynolds number viscous flow above body surface, as shown in Figure 4, in the region very close to the surface, especially in the sublayer region of the boundary layer, regular tri-linear interpolation method cannot give satisfied results, because the value of boundary point of the dashed line grid is interpolated from the solutions of the solid line grids, which are not at the same distance from the surface. This is because for high Reynolds number problems, the velocity profiles have very high gradient in the surface normal direction; a small difference in the surface normal direction can bring out a big difference in velocity value, so the normal interpolation method needs to be modified. Therefore, it is necessary to develop new, more efficient methods and logic, which can handle more general and complex problems, such as viscous flow problems.

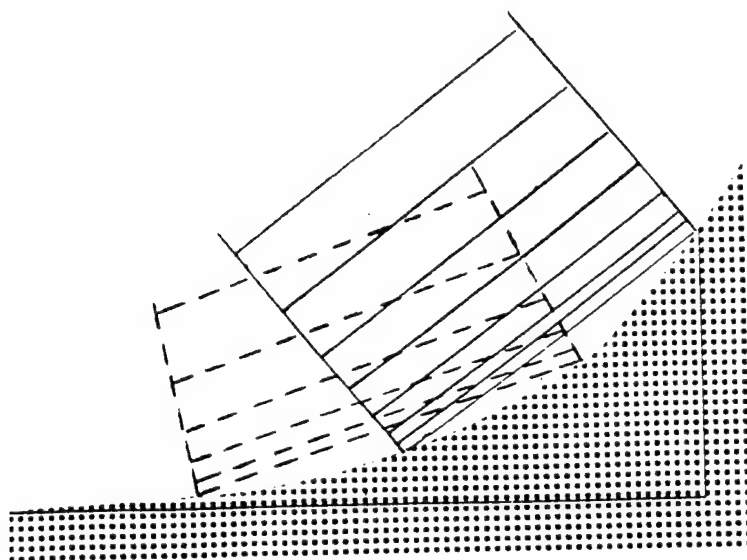


Figure 3. Concave surface grids

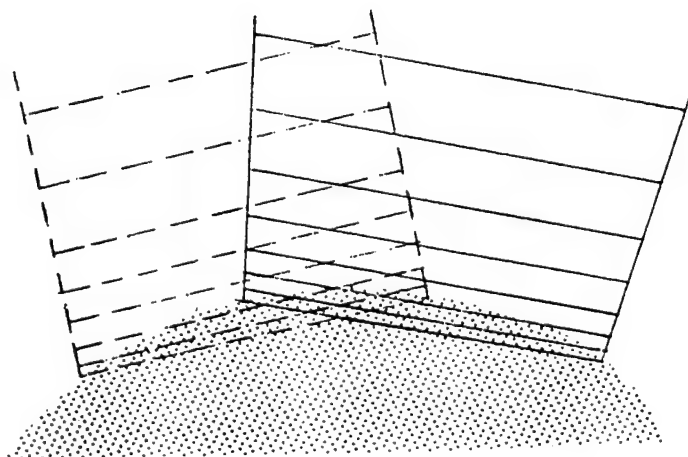


Figure 4. High Reynolds number boundary layer flow

## 2.2 Chimera Method

The grid embedding has two major parts: decomposition of the domain into overlapping subdomains and communication among the subdomains. Each subdomain is chosen to lessen the effort of generating an acceptable mesh, each of these meshes represents a separate and independent grid generation. To increase the flexibility in the domain decomposition, it is necessary to remove the regions of a mesh containing an embedded grid from that mesh, that is, an embedded mesh introduces a 'hole' into the mesh in which it is embedded. Because appropriate boundary values are interpolated from the mesh in which the boundary is embedded, and these parts do not enter into the solution process, inter-grid communication is simplified through the grid boundaries. Therefore, the Chimera procedure naturally separates into two parts: (1) generation of the composite mesh and associated interpolation data and (2) solution of the flow equations on the composite and interpolation data mesh. Each part is embodied in a separate computer code. The first code takes the independently generated composite mesh as input and automatically constructs the composite mesh as an output, which requires establishment of the proper lines of communication among the grids through tri-linear interpolations within the overlapping regions to determine outer boundary point properties for each grid. In addition to the outer boundary point communication, holes arise whenever the solid surfaces contained by one grid are also contained within other grids, and the hole requires boundary condition; therefore, it is necessary to identify points and locate points from which boundary values can be interpolated, then evaluate interpolation parameters. Because the hole point and hole boundaries point are not at the computation domain, they are imposed by blanking out the control equations within the solution matrix and directly substituting the boundary condition. A detailed discussion of the methodologies used to tag hole boundary points and outer boundary points is given in following sections 2.2 and 2.3.

Once the subgrids were constructed, and the interpolation coefficients were computed and stored by the assembly code described above, the second code can use the stored interpolation coefficients to provide boundary condition for each grid, and solve the appropriate flow equation on each grid with single grid solver. A detailed discussion is given in Section 2.4.

### 2.2.1 Hole Creation

Because each component mesh is generated independently, complications frequently arise when the grids are embedded. For example, points of an enclosing mesh  $G_{l,i}$ , may be found to lie within a solid boundary contained within an embedded grid  $G_{l+1,i}$ , such points lie within the solid boundary and outside the computational domain, therefore, must be excluded from the solution process. Thus, the embedded mesh  $G_{l+1,i}$  introduces a "hole" in  $G_{l,i}$ . The only computational requirement is that there remain a sufficient overlap to support an interpolation for the outer boundary of  $G_{l+1,j}$  from points in  $G_{l,i}$ .

There is a two-step procedure in finding the hole points as well as the hole boundary. First, (1) find a geometric center of embedded body surface as a temporary origin, say  $P_0$  (2) find the  $R_{\max}$  and  $R_{\min}$  as the maximum and minimum distances from  $P_0$  to points on embedded body surface. All the points that fall outside of the larger sphere with radial of  $R_{\max}$  are considered to be field points. All the points that fall inside of a smaller sphere with radial of  $R_{\min}$  are considered to be "hole" points, those points that fall between the two spheres need a more accurate method to tell whether or not a point is a hole point. To avoid unnecessary confusion in a three-dimensional case, a two-dimensional case is presented to illustrate the idea of the hole searching procedure. In Figure 5, construct outward normal to the initial hole boundary at each point defining the surface,  $C$ . Test the magnitude of  $\vec{r}$ , the position vector relative to  $P_0$ . For every point  $P$

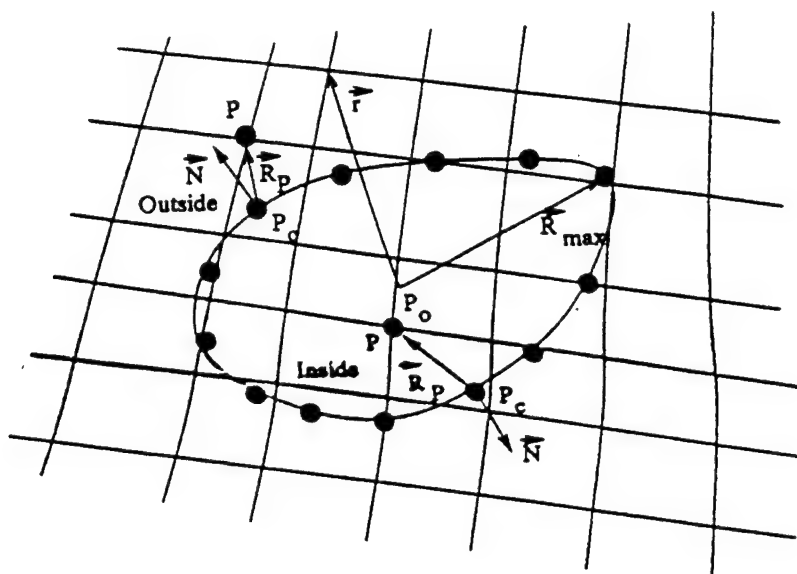


Figure 5. Normal vector to the initial hole boundary

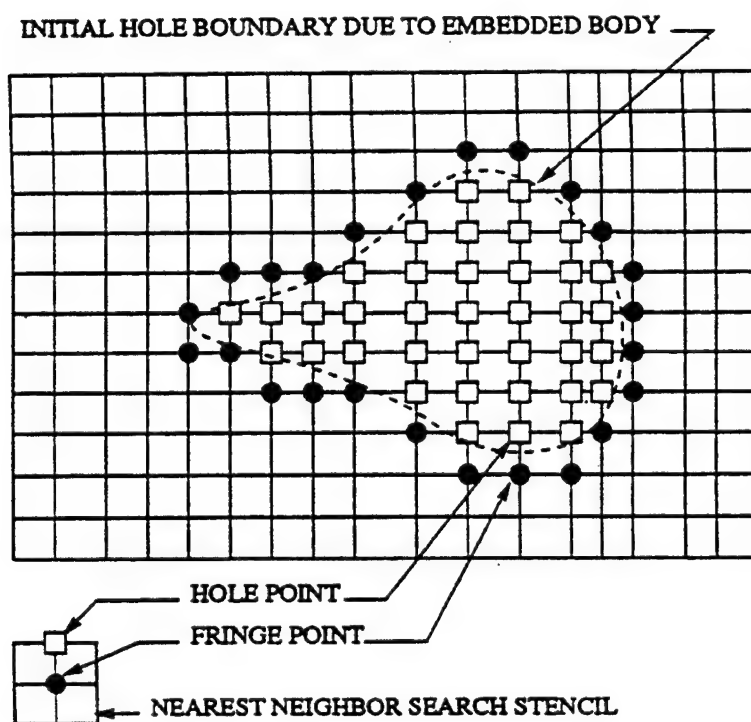


Figure 6. Construction of fringe points

of  $G_{i,i}$ ; if the  $|\vec{r}| < R_{\max}$ ,  $P$  may be inside  $C$ . So, form  $\vec{N} \cdot \vec{R}_p$  where  $\vec{N}$  is the unit normal at the surface point,  $P_C$ , the closet point to  $P$ , and  $\vec{R}_p$  is the position vector of  $P$  relative to  $P_C$ ; if  $\vec{N} \cdot \vec{R}_p < 0$ ,  $P$  lies within the hole and the flag variable, IBLANK, is set to zero, otherwise,  $P$  is outside the hole and IBLANK is set to 1. The IBLANK variable is used by the flow solver to determine whether a point should enter into the solution process or not, as illustrated by the equation below:

$$A\Delta Q = \text{IBLANK} * \text{RHS}$$

where  $A$  is the coefficient matrix,  $\Delta Q$  is the change in flow solution, and RHS is the source term. Thus for the hole points, the above equation reduces to

$$\Delta Q = 0$$

and the values of the variable at the hole points are not changed in the solution process.

The points of  $G_{i,i}$  within a hole are excluded from the solution and are not usable as boundary points. Therefore, additional points of  $G_{i,i}$  are identified as hole boundary or fringe points. After the hole points are found, the hole boundary points or fringe points can easily be located by searching the IBLANK values of neighboring points. Figure 6 illustrates the construction of fringe points. The procedure is to examine the search stencil for each point,  $P$ , in  $G_{i,i}$  at which IBLANK=1; if one of the nearest neighbors is a hole points,  $P$  is a fringe point. The indices of the fringe points are added to a list of boundary points which require interpolation data. The values of the unknowns at these boundary points are interpolated from the embedded mesh,  $G_{i,i+1}$ .

### 2.2.2 Searching for Interpolation Elements

Communication among the grid is achieved by interpolation of boundary values from the grids in which the boundaries are contained. Care must be taken in the application of the method to ensure that the interpolated point lies within this cell.

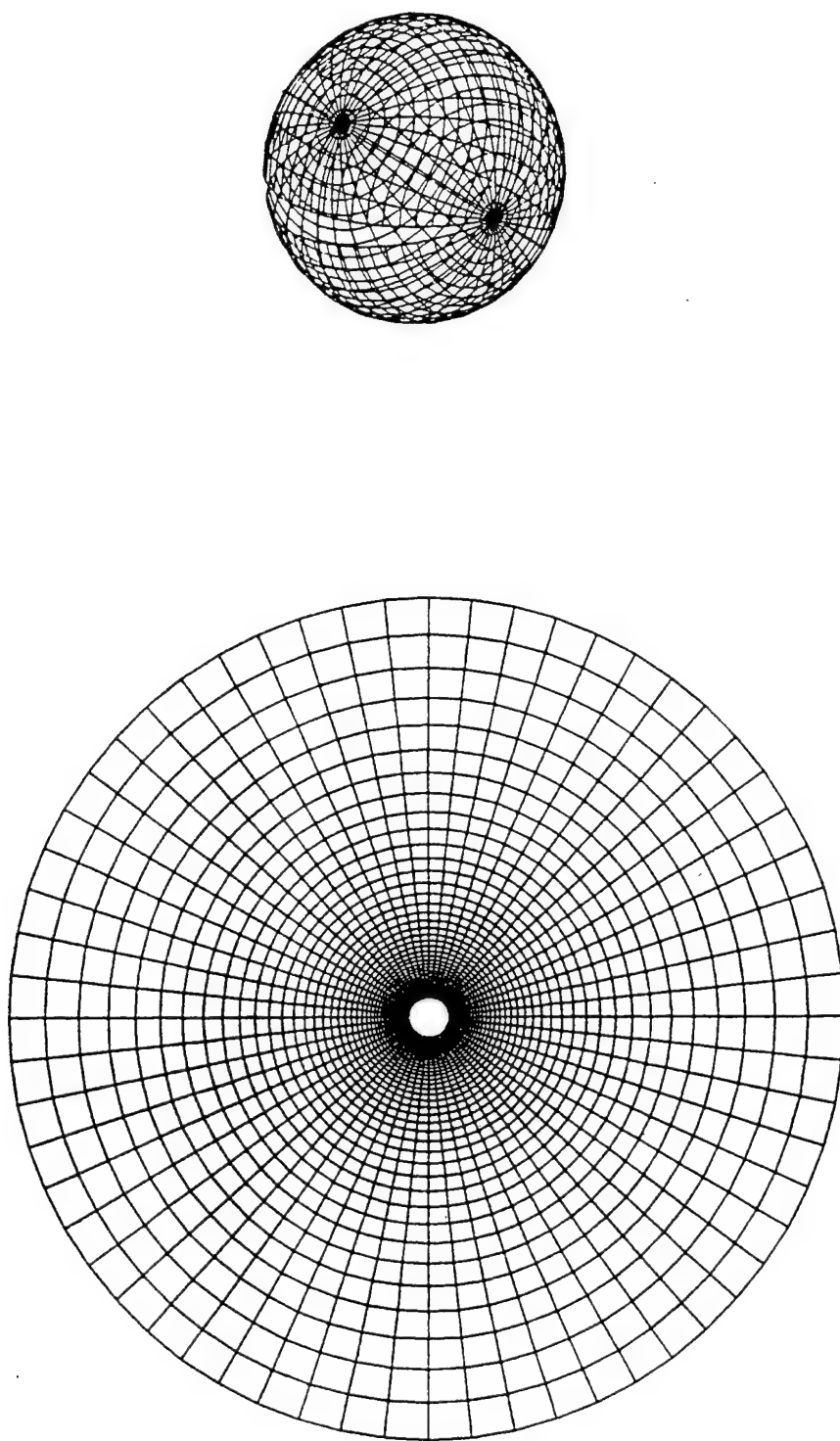


Figure 7. A sphere surface covered by a single piece of grid.

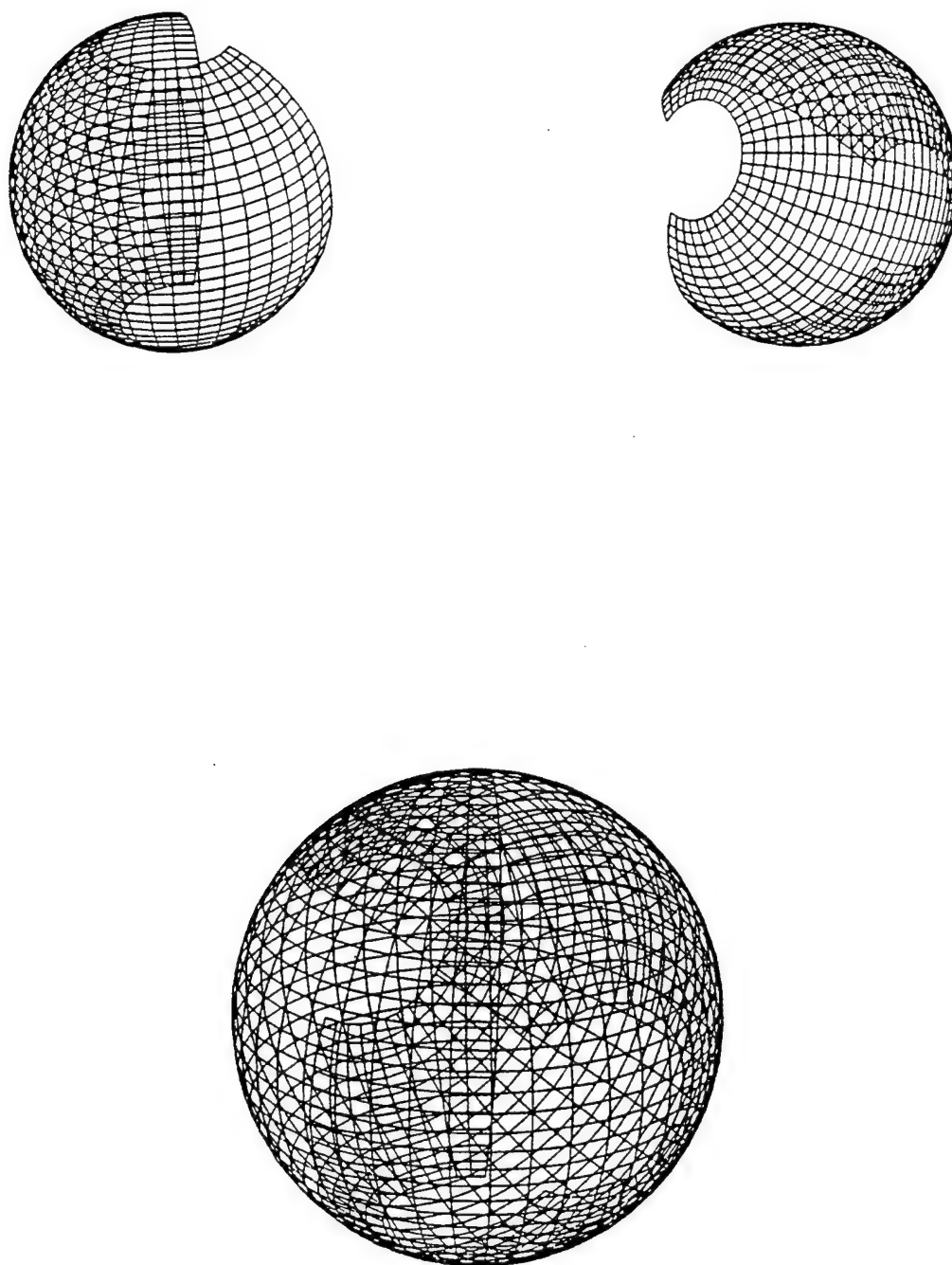


Figure 8. The whole sphere surface covered by two pieces of grids.

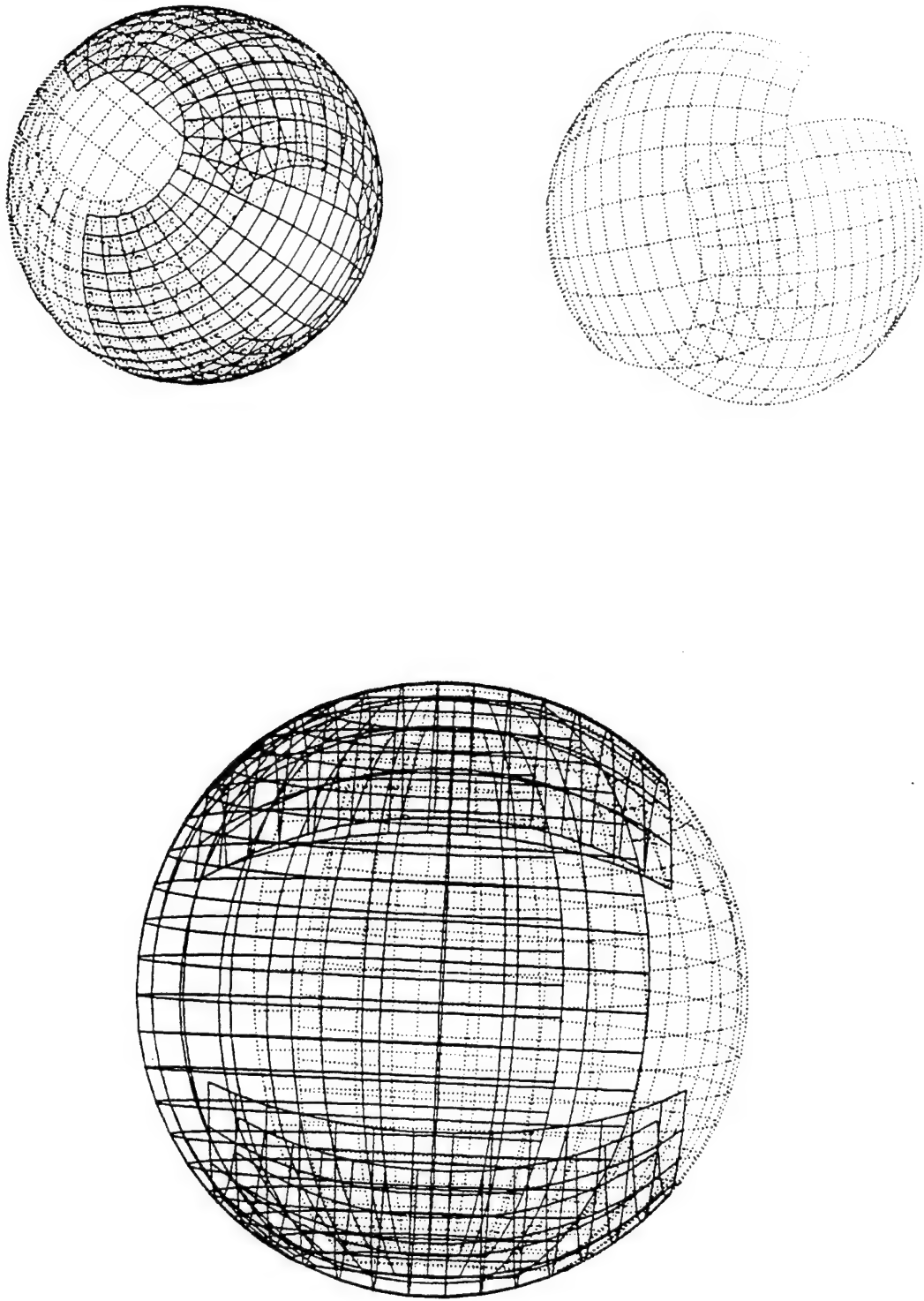


Figure 9. The whole sphere surface covered by four pieces of grids.

For the overset grid schemes interpolation is generally needed to update boundary points, and interpolation requires routines to seek out nearest points (or cells) to interpolate. Finding these points can be costly and prone to error. Moreover, as Figures 7, 8 and 9 indicate, even for the nearest points in space, it is not necessarily that the best interpolation results are obtained.

There are two simple methods which can be used to find the cell which contains given boundary grid. One of the methods is developed in the physical coordinate  $x, y, z$  space, which is called "stencil" method. The other method is developed in the computational coordinate  $\xi, \eta, \zeta$  space, which is called Jacobin matrix method, because it needs to use Jacobin matrix to calculate the interpolation coefficients. For our case, the Jacobin matrix method is more efficient than "stencil" method, because the Jacobin matrix was calculated in other subroutines in the same program.

For more than two mesh problems, it is possible that the boundary of each mesh is overlapped by other two or more meshes, which means for the same point of a given boundary, its values can be interpolated from different meshes, therefore, it is necessary to introduce new logic and a series of parameters to judge for which grids the interpolated boundary data are best; therefore, the data structure becomes very complicated. On the other hand, the simple logic can fail because of the need to introduce a starting point, which is not far from the destination point and in the same grid. For the general multiple overset grids, these conditions are not always satisfied.

### **2.2.3 Interpolation Points**

Because the separate grids are to be treated as independent entities, boundary conditions must be supplied to each. The boundary conditions of the differential equations which model the flow provide data only at the boundaries of the computational domain. Thus, other data must be obtained for the subdomain boundaries which are not coincident with those of the computational domain. Because the subdomain boundaries typically lie in

the interior of the computational domain where the differential equations are valid, it seems appropriate that the solution of these equations should provide the necessary boundary data. Appropriate boundary values are interpolated from the mesh or meshes in which the boundary is embedded. Thus, it is necessary to find the interpolation points or cells on the embedded mesh from which the boundary values are interpolated. Once these points or cells are found, their pointers are added to a list of such points to be used by the flow solver to update the variables at the hole or boundary points. There are several different procedures to locate the points in the embedded mesh which would be best for interpolation boundary point. In general, the closet points in the embedded mesh would provide the best interpolation data for the boundary points, but for some special cases, it is necessary to pay special attention in selecting the interpolation points. For viscous flow problems, in the region very near the surface, the closest points would not necessarily provide the best interpolation data, because the flow properties change very fast in the surface normal direction. A small distance in this direction would bring a big change in the flow properties. Therefore, we have to find those points with the same height as the hole or boundary points, and use them as interpolation points. Because these points may not be a real grid point in the physical space, it is necessary to find a method to use the results of these real grid points to get the interpolation data.

#### **2.2.4 Interpolation Methods**

Because the subdomain boundary is in the interior of other computational domain where the differential equations are valid, it is appropriate to use the solution of these equations to update the necessary boundary data. There are several approaches to obtain these boundary data, and all of them involve some form of data interpolation from one mesh to another mesh.

We can use tri-linear interpolation to update the values of boundary points for a three-dimensional problem. The tri-linear interpolation which has the form:

$$A = \alpha_1 + \alpha_2 \bar{\xi} + \alpha_3 \bar{\eta} + \alpha_4 \bar{\zeta} + \alpha_5 \bar{\zeta} \bar{\eta} + \alpha_6 \bar{\eta} \bar{\xi} + \alpha_7 \bar{\xi} \bar{\zeta} + \alpha_8 \bar{\xi} \bar{\eta} \bar{\zeta}$$

where  $0 < \bar{\xi}, \bar{\eta}, \bar{\zeta} < 1$  are the coordinates of the point to be interpolated and the  $\alpha_1 - \alpha_8$  depend upon the function values at the points forming the interpolation cell.

In general, the tri-linear interpolation can give good interpolated results. But in the region very near the body surface with very fine grids in the normal direction, this method may give wrong results because of geometric and physical reasons. Because different interpolation methods need different kinds of data structure, it is necessary to transfer different kinds of data structure into a standard data structure for easy data management.

#### 2.2.5 Data Structure

A Chimera method requires the management of a large amount of grids, solution data and interpolation relationships, which consist of a list of every boundary point in each grid which receives interpolated information from other grids, a list of every grid element in each grid which provides interpolated information to boundary points, a list of all points in the composite grid that are hole and boundary (i.e., "blanked") points, and the pointer from boundary points to the grid elements which provide the boundary point with interpolated solution data. Additional information, such as grid-storage location, number of points in each coordinate direction, location of interpolation stencils, location of interpolation coefficients, location of interpolated boundary data, and location of hole or excluded points, should also be included. Therefore, it is necessary to keep track of the storage locations of the coordinates of each grid, solution data on each grid, all of the interpolation information, and develop a good management method to transfer all these data to the user in an automatic manner.

Because most of these data are stored in lists, connections among the data are made through the use of linked lists and pointer. It is necessary to store the position of the first element (a pointer) in each grid and the number of points in each coordinate direction on each grid to locate the coordinates of any points.

### 2.3 Grid Generation and Composite Grid

Although the use of overset-grid approach reduces the complexity of grid generations, the task of generating a satisfactory body conforming grid about a three-dimensional complex geometric body is still not easy. In general, the grids must not be too distorted. They should have smooth variation, and should be clustered to flow field regions, typically near boundary surfaces. Moreover, the grids should be generated in an automatic manner that requires a minimum of user input.

Because of the application of boundary conditions can be simplified in finite difference calculation if grid lines coincide with boundary lines, body conforming curvilinear grids are often used in finite different flow field simulation. One approach for generating body conforming grid is to solve a set of partial differential equations with a series of monotone variation level lines  $\xi(x,y,z)$ ,  $\eta(x,y,z)$  and  $\zeta(x,y,z)$  as a solution of these partial differential equations. The values of these variable  $\xi, \eta$ , and  $\zeta$  are user-specified on the boundary surface, and constraints expressed as differential equations are used to develop the grid away from the boundaries. The most popular approaches require the solution of a set of elliptic equations which satisfies the maximum principle. For external aerodynamics applications, the placement of the outer boundary curve will not have to be precisely specified, especially for Chimera overset grid approach. Solving a set of hyperbolic or parabolic governing equations is, however, more powerful and efficient.

The governing grid generation equations are determined by the differential constraints of two mesh line orthogonality relations between  $\xi$  and  $\eta$  and between  $\eta$  and  $\zeta$ ,

$$\bar{r}_\xi \cdot \bar{r}_\eta = 0$$

$$\bar{r}_\eta \cdot \bar{r}_\zeta = 0$$

and a user specified mesh cell size constraint.

$$\left| \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} \right| = \Delta V$$

where the  $\bar{r}$  is the position vector,  $(x, y, z)$ . The grid is obtained by first defining a surface grid, then the governing equations have to be solved by marching outward in the normal direction from specified distribution of points on the inner body boundary, under the constraint of the two orthogonal relations and the user specified spacings or volumes. Details of the hyperbolic grid generation procedure can be found in Ref. 55.

In the overset mesh technique a major grid stretches over the entire field. The major grid might be a simple rectangular grid or a grid generated about a dominant boundary or body surface. Minor grids are generated about remaining portions of the body-configuration or any other special feature such as an intense vorticity region. The minor grids are used to resolve features of the geometry or flow that are not adequately resolved by the major grid. They are generated somewhat independently of the major grid and are overset on top of the major grid without requiring mesh boundaries to joint in any special way. Overall, the effect of one grid is imposed upon the other by means of boundary data that is interpolated back and forth as the iterative solution process of solving the difference equations on each grid proceeds. The minor grid outer boundary data comes from the major grid solution, not from a far field boundary condition routine.

As the separate grids are to be treated as independent entities, boundary conditions must be supplied to each. The boundary conditions of the differential equations which model the flow provide data only at the boundaries of the computational domain. Thus, other data must be obtained for the subdomain boundaries which are not coincident with those of the computational domain. Because the subdomain boundaries typically lie in the interior of the computational domain where the differential equations are valid, it seems appropriate that the solution of these equations should provide the necessary boundary data. Appropriate boundary values are interpolated from the mesh or meshes in which the boundary is embedded.

Finding a grid element which can provide interpolated information to a hole boundary point is a necessary but not sufficient condition for an interpolation to be considered satisfactory. For example, a grid which is to supply interpolated information to a boundary point may itself be boundary point which is updated through interpolation. Most of the information used to update the original boundary point will come from interpolation rather than directly from the flow solution. This close coupling of boundary points can result in poor global convergence. Therefore, in the grid generation and composite grids, a sufficient overlap region (at least two points) must be provided and in Chimera connecting code, a simple strategy is added to not allow boundary points to be updated by grid elements which are themselves comprised of boundary points.

## **2.4 Solution Procedure**

The Chimera approach has the advantage that a flow simulation code developed for a single general curvilinear grid can be readily adapted for composite overset grids. It is easy to write an outer loop control program that calls in grids and interface routines and provides logic to set communication relations between the grids. At each time step, a grid and its data are fetched from an isolated memory device into the working memory.

Boundary interface arrays that store grid interconnect data,  $Q_{BC}$ , are also brought in. The  $Q_{BC}$  arrays hold overset-grid boundary values for the current grid which are supplied from the other grids. The solution on the current grid is then updated or advanced in time using the flow algorithm, the  $Q_{BC}$  interface boundary values, and the usual boundary condition routines (tangency) that are applicable. Overset boundary data that the current grid sends to other grids are then loaded into  $Q_{BC}$  and all the arrays are sent back to the isolated large memory. The next grid is then fetched and so on.

Because of the above reasons, the intention is to build a simple solver which can be used for quick embedment within the Chimera framework, check outputs, give good guessing, diagnostics, etc., as well as for use with an optional fortified approach. These numerical algorithms for inviscid flow simulation are efficient, and simple and robust so as to minimize the amount of engineering man hours to obtain an initial solution. Therefore, a formulation for the steady inviscid flow, in terms of primitive variables of the original Cartesian velocity variables is described for body fitted general curvilinear coordinates. The details of the formulation are given in the later chapter. A limited number of calculations have been undertaken to validate the formulation.

### 3.0 FLOW SOLVER

#### 3.1 Subsonic Steady Inviscid Nonconservative Formulation

*Continuity:*

$$u_x + v_y + w_z = -\vartheta$$

with

$$\vartheta = (u\rho_x + v\rho_y + w\rho_z)/\rho$$

*Vorticity Definitions:*

$$\omega_1 = w_y - v_z$$

$$\omega_2 = u_z - w_x$$

$$\omega_3 = v_x - u_y$$

Written as a system of first order equations

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_y \\ v_y \\ w_y \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_z \\ v_z \\ w_z \end{pmatrix} = \begin{pmatrix} -\vartheta \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (1)$$

or

$$\tilde{A}\tilde{q}_x + \tilde{B}\tilde{q}_y + \tilde{C}\tilde{q}_z = \tilde{g} \quad (2)$$

Multiply by  $D_1\tilde{A}^t$ , with  $D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$  then

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_y \\ v_y \\ w_y \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_z \\ v_z \\ w_z \end{pmatrix} = \begin{pmatrix} -\vartheta \\ -\omega_3 \\ \omega_2 \end{pmatrix} \quad (3)$$

or

$$A_1 \vec{q}_x + B_1 \vec{q}_y + C_1 \vec{q}_z = \vec{f}_1$$

Each matrix.  $A_1$ ,  $B_1$ , and  $C_1$  has real eigenvalues. Moreover, the matrix  $aA_1 + bB_1 + cC_1$  has **distinct real eigenvalues**

$$\lambda(aA_1 + bB_1 + cC_1) = -a, \pm \sqrt{a^2 + b^2 + c^2}$$

and a complete set of eigenvectors must exist (Mathematica could not find them. so they will need to be worked out ) so the formulation is applicable to general coordinates as will be noted later.

If Eq.(1) is multiplied by  $D_2 \tilde{B}^t$ , with  $D_2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$  then

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} u_y \\ v_y \\ w_y \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_z \\ v_z \\ w_z \end{pmatrix} = \begin{pmatrix} \omega_3 \\ -\vartheta \\ -\omega_1 \end{pmatrix} \quad (4)$$

and Eq.(4) is similar to Eq.(3).

If Eq.(1) is multiplied by  $D_3 \tilde{C}^t$ , with  $D_3 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$  then

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_y \\ v_y \\ w_y \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_z \\ v_z \\ w_z \end{pmatrix} = \begin{pmatrix} -\omega_2 \\ \omega_1 \\ -\vartheta \end{pmatrix} \quad (5)$$

For symmetry, add Eqs.(3), (4) and (5)

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} u_x \\ v_x \\ w_x \end{pmatrix} + \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} u_y \\ v_y \\ w_y \end{pmatrix} + \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} u_z \\ v_z \\ w_z \end{pmatrix} \\ = \begin{pmatrix} \omega_3 - \omega_2 - \vartheta \\ \omega_1 - \omega_3 - \vartheta \\ \omega_2 - \omega_1 - \vartheta \end{pmatrix} \quad (6)$$

or

$$A\vec{q}_x + B\vec{q}_y + C\vec{q}_z = \vec{f} \quad (7)$$

Each coefficient matrix has real distinct eigenvalues and the linear combination  $aA + bB + cC$  has the distinct real eigenvalues

$$\lambda(aA + bB + cC) = -a - b - c, \pm \sqrt{3(a^2 + b^2 + c^2)} \quad (8)$$

and again a complete set of eigenvectors must exist.

### 3.2 General Coordinates

Let  $\xi = \xi(x, y, z)$ ,  $\eta = \eta(x, y, z)$ , and  $\zeta = \zeta(x, y, z)$  then Eq.(7) becomes

$$(\xi_x A + \xi_y B + \xi_z C)\vec{q}_\xi + (\eta_x A + \eta_y B + \eta_z C)\vec{q}_\eta + (\zeta_x A + \zeta_y B + \zeta_z C)\vec{q}_\zeta = \vec{f} \quad (9)$$

All combinations of eigenvalues follow from Eq.(8).

Rewrite (9) as

$$\hat{A}\vec{q}_\xi + \hat{B}\vec{q}_\eta + \hat{C}\vec{q}_\zeta = \vec{f} \quad (10)$$

where

$$\hat{A} = \begin{pmatrix} \xi_x - \xi_y - \xi_z & \xi_x + \xi_y & \xi_x + \xi_z \\ \xi_x + \xi_y & -\xi_x + \xi_y - \xi_z & \xi_y + \xi_z \\ \xi_x + \xi_z & \xi_y + \xi_z & -\xi_x - \xi_y + \xi_z \end{pmatrix}$$

$$\hat{B} = \begin{pmatrix} \eta_x - \eta_y - \eta_z & \eta_x + \eta_y & \eta_x + \eta_z \\ \eta_x + \eta_y & -\eta_x + \eta_y - \eta_z & \eta_y + \eta_z \\ \eta_x + \eta_z & \eta_y + \eta_z & -\eta_x - \eta_y + \eta_z \end{pmatrix}$$

$$\hat{C} = \begin{pmatrix} \zeta_x - \zeta_y - \zeta_z & \zeta_x + \zeta_y & \zeta_x + \zeta_z \\ \zeta_x + \zeta_y & -\zeta_x + \zeta_y - \zeta_z & \zeta_y + \zeta_z \\ \zeta_x + \zeta_z & \zeta_y + \zeta_z & -\zeta_x - \zeta_y + \zeta_z \end{pmatrix}$$

and

$$\vec{f} = \begin{pmatrix} \omega_3 - \omega_2 - \vartheta \\ \omega_1 - \omega_3 - \vartheta \\ \omega_2 - \omega_1 - \vartheta \end{pmatrix}$$

Here

$$\omega_1 = (\xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\zeta) - (\xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta)$$

$$\omega_2 = (\xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta) - (\xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta)$$

$$\omega_3 = (\xi_x v_\xi + \eta_x v_\eta + \zeta_x v_\zeta) - (\xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta)$$

$$\vartheta = \frac{(U\rho_\xi + V\rho_\eta + W\rho_\zeta)}{\rho}$$

$$U = \xi_x u + \xi_y v + \xi_z w, \quad V = \eta_x u + \eta_y v + \eta_z w, \quad W = \zeta_x u + \zeta_y v + \zeta_z w$$

### 3.3 Numerical Algorithm

Euler Implicit ( $h = \Delta t$  or relaxation factor):

$$(I + h\hat{A}^n\delta_\xi + h\hat{B}^n\delta_\eta + h\hat{C}^n\delta_\zeta)\Delta\bar{q}^n = -h[(\hat{A}^n\delta_\xi + \hat{B}^n\delta_\eta + \hat{C}^n\delta_\zeta)\bar{q}^n - \bar{f}^n] \quad (11)$$

where  $\Delta\bar{q}^n = \bar{q}^{n+1} - \bar{q}^n$

Approximately factored in ADI-like fashion

$$(I + h\hat{A}^n\delta_\xi)(I + h\hat{B}^n\delta_\eta)(I + h\hat{C}^n\delta_\zeta)\Delta\bar{q}^n = -h[(\hat{A}^n\delta_\xi + \hat{B}^n\delta_\eta + \hat{C}^n\delta_\zeta)\bar{q}^n - \bar{f}^n] \quad (12)$$

Algorithm form:

$$(I + h\hat{A}^n\delta_\xi)\Delta\bar{q}^* = -h[(\hat{A}^n\delta_\xi + \hat{B}^n\delta_\eta + \hat{C}^n\delta_\zeta)\bar{q}^n - \bar{f}^n]$$

$$(I + h\hat{B}^n\delta_\eta)\Delta\bar{q}^{**} = \Delta\bar{q}^*$$

$$(I + h\hat{C}^n\delta_\zeta)\Delta\bar{q}^n = \Delta\bar{q}^{**} \quad (13)$$

$$\bar{q}^{n+1} = \bar{q}^n + \Delta\bar{q}^n$$

With smoothing, above equations become:

$$(I + h\hat{A}^n\delta_\xi - h\epsilon_i(\Delta\nabla)_\xi)\Delta\bar{q}^* = -h[(\hat{A}^n\delta_\xi + \hat{B}^n\delta_\eta + \hat{C}^n\delta_\zeta)\bar{q}^n - \bar{f}^n] - h\epsilon_e[(\Delta\nabla)_\xi^2 + (\Delta\nabla)_\eta^2 + (\Delta\nabla)_\zeta^2]\bar{q} \quad (14)$$

$$(I + h\hat{B}^n\delta_\eta - h\epsilon_i(\Delta\nabla)_\eta)\Delta\bar{q}^{**} = \Delta\bar{q}^*$$

$$(I + h\hat{C}^n\delta_\zeta - h\epsilon_i(\Delta\nabla)_\zeta)\Delta\bar{q}^n = \Delta\bar{q}^{**}$$

$$\bar{q}^{n+1} = \bar{q}^n + \Delta \bar{q}^n$$

The left hand side of first 3 equations of (14) can be written as:

$$[I + h\hat{A}^n \delta_\xi - h\epsilon_i(\Delta \nabla)_\xi] \Delta \bar{q}^* = [(-h\frac{\hat{A}}{2} - h\epsilon_i), (I + 2\epsilon_i h), (h\frac{\hat{A}}{2} - h\epsilon_i)] \Delta \bar{q}^*$$

$$[I + h\hat{A}^n \delta_\eta - h\epsilon_i(\Delta \nabla)_\eta] \Delta \bar{q}^{**} = [(-h\frac{\hat{B}}{2} - h\epsilon_i), (I + 2\epsilon_i h), (h\frac{\hat{B}}{2} - h\epsilon_i)] \Delta \bar{q}^{**}$$

$$[I + h\hat{A}^n \delta_\zeta - h\epsilon_i(\Delta \nabla)_\zeta] \Delta \bar{q}^n = [(-h\frac{\hat{C}}{2} - h\epsilon_i), (I + 2\epsilon_i h), (h\frac{\hat{C}}{2} - h\epsilon_i)] \Delta \bar{q}^n$$

### 3.4 Bernoulli Equation

If the flow is steady, inviscid, adiabatic, the Bernoulli equation can be derived from the Crocco relation and perfect gas relations (Anderson et al., 1984):

$$\frac{\rho}{\rho_\infty} = \left[ 1 + \frac{\gamma - 1}{2} \left( M_\infty^2 - \frac{u^2 + v^2 + w^2}{a_\infty^2} \right) \right]^{\frac{1}{\gamma - 1}} e^{-(s - s_\infty)/R}$$

For irrotational flow, the entropy correction term,  $e^{-(s - s_\infty)/R}$ , can be dropped. To avoid computational cost this is expanded using the binomial expansion:

$$\frac{\rho}{\rho_\infty} = (1 \pm \alpha)^n = 1 \pm \alpha n + \frac{n(n-1)}{2} \alpha^2 \pm \frac{n(n-1)(n-2)}{2 \times 3} \alpha^3 + \dots$$

where

$$n = 1/(\gamma - 1)$$

$$\alpha = \frac{\gamma - 1}{2} \left( M_\infty^2 - \frac{u^2 + v^2 + w^2}{a_\infty^2} \right)$$

Written as

$$(1 + \alpha)^n = 1 + \alpha n \left( 1 + \alpha \frac{(n-1)}{2} \left( 1 + \alpha \frac{(n-2)}{3} \left( 1 + \alpha \frac{(n-3)}{4} \dots \right) \right) \right)$$

or

$$(1 + \alpha)^n = 1 + \alpha \left( n + \alpha \left( \frac{n(n-1)}{2} + \alpha \left( \frac{n(n-1)(n-2)}{6} + \alpha \frac{n(n-1)(n-2)(n-3)}{12} \right) \right) \right)$$

or

$$(1 + \alpha)^n = 1 + \alpha(c_1 + \alpha(c_2 + \alpha(c_3 + \alpha c_4)))$$

where

$$\begin{aligned} c_1 &= \frac{1}{\gamma - 1} \\ c_2 &= \frac{c_1}{2} \left( \frac{1}{\gamma - 1} - 1 \right) \\ c_3 &= \frac{c_2}{3} \left( \frac{1}{\gamma - 1} - 2 \right) \\ c_4 &= \frac{c_3}{4} \left( \frac{1}{\gamma - 1} - 3 \right) \end{aligned}$$

Therefore

$$\frac{\rho}{\rho_\infty} = 1 + \alpha(c_1 + \alpha(c_2 + \alpha(c_3 + \alpha c_4)))$$

### 3.5 Boundary Conditions

Boundary conditions are taken from a vorticity/tangency inviscid relation.

From the vorticity definitions:

$$\zeta_y w_\zeta - \zeta_z v_\zeta = g_1 = \omega_1 - (\xi_y w_\xi + \eta_y w_\eta - \xi_z v_\xi - \eta_z v_\eta)$$

$$\zeta_z u_\zeta - \zeta_x w_\zeta = g_2 = \omega_2 - (\xi_z u_\xi + \eta_z u_\eta - \xi_x w_\xi - \eta_x w_\eta)$$

$$\zeta_x v_\zeta - \zeta_y u_\zeta = g_3 = \omega_3 - (\xi_x v_\xi + \eta_x v_\eta - \xi_y u_\xi - \eta_y u_\eta)$$

On the wall, these equations can be written as

$$\zeta_y w - \zeta_z v = -\Delta\zeta g_1 + \zeta_y w^* - \zeta_z v^* \equiv \hat{g}_1$$

$$\zeta_z u - \zeta_x w = -\Delta\zeta g_2 + \zeta_z u^* - \zeta_x w^* \equiv \hat{g}_2$$

$$\zeta_x v - \zeta_y u = -\Delta\zeta g_3 + \zeta_x v^* - \zeta_y u^* \equiv \hat{g}_3$$

where  $u^*$ ,  $v^*$  and  $w^*$  are values of  $u$ ,  $v$ , and  $w$  at  $\Delta\zeta$  above the wall.

The condition on the surface  $\zeta = 0$  is given by

$$\zeta_x u + \zeta_y v + \zeta_z w = 0$$

Solving these four equations for the three unknowns  $u$ ,  $v$ , and  $w$  on the surface by generalized inverse (or using the vorticity relations to remove the other components from the tangency relation) gives the vorticity/tangency relations:

$$(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)u = \zeta_z \hat{g}_2 - \zeta_y \hat{g}_3$$

$$(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)v = \zeta_x \hat{g}_3 - \zeta_z \hat{g}_1$$

$$(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)w = \zeta_y \hat{g}_1 - \zeta_x \hat{g}_2$$

or in vector form

$$(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)\vec{q} = -\nabla\zeta \times \vec{\hat{g}}$$

## 4.0 VERIFICATION AND APPLICATION

### 4.1 Potential Solution for Flow around a Sphere

The ideal flow over a sphere is given by the superposition of a uniform stream and a doublet. The stream function of these contributions can be written as:

$$\psi = \frac{1}{2}r_0^2 U \sin^2 \theta \left[ \left( \frac{r}{r_0} \right)^2 - \left( \frac{r}{r_0} \right)^{-1} \right]$$

in which  $r_0$  is radius of the sphere, and it is easy to verify that for arbitrary  $\theta$ ,  $r = r_0$  is a spherical stream surface where  $\psi = 0$ . The velocity components for this flow are:

$$V_r = U \cos \theta \left[ 1 - \left( \frac{r}{r_0} \right)^3 \right]$$
$$V_\theta = -\frac{1}{2} U \sin \theta \left[ 2 + \left( \frac{r}{r_0} \right)^3 \right]$$

on the surface of the sphere the velocity is

$$q = V_\theta = -\frac{3}{2} U \sin \theta$$

The maximum occurs at the equator, where  $q = -3U/2$ . The surface pressures are found by Bernoulli's equation on the surface of the sphere, in terms of the pressure coefficient, the result is

$$C_p = 1 - \frac{9}{4} \sin^2 \theta$$

We can transfer the solutions  $(V_r, V_\theta)$  in spherical coordinate system to  $(u, v, w)$  of Cartesian coordinate system with following relations:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\cos\theta = \frac{x}{r}$$

$$\sin\theta = \sqrt{1 - \cos^2\theta}$$

$$\cos\psi = \frac{y}{r\sin\theta}$$

$$\sin\psi = \frac{z}{r\sin\theta}$$

then:

$$u = V_r \cos\theta + V_\theta \sin\psi$$

$$v = V_r \sin\theta \cos\psi + V_\theta \cos\theta \cos\psi$$

$$w = V_r \sin\theta \sin\psi + V_\theta \cos\theta \sin\psi$$

With these relations, for any given grid point position  $(x, y, z)$ , it is easy to have the potential solution  $u, v$ , and  $w$ .

## 4.2 Solutions for Flow around a Sphere in Single Grid

Because the analytical solution for a potential flow around a sphere is available for comparison, and the sphere grid is relatively easy to generate, the flow pass a sphere was selected as the model to verify the simple inviscid flow solver given above. Theoretically, the flow solver given above should give the same result as

potential flow solution as long as the vorticity terms disappear from the source terms in equation (1). The Mach number contours on sphere surface and a cross section in computational space are given in Figure 10 through 13. All of these results show good agreement with the analytical results, therefore, it was concluded that the codes were functioning properly.

### 4.3 Flow Solutions with Multioverset Grids

Since the simple inviscid flow solver and connecting codes are working, it is necessary to test these codes by calculating relative complex geometric configurations. We have a hyperbolic surface grid generation program and the 3-D hyperbolic grid generation program. For a complex multibody configuration, it is not too difficult to use these grid generators to generate 3-D grid on each body, and superimpose them together, and use the Chimera method to solve the whole problem. Even just using the simple inviscid flow solver, it still can get some good approximate results which would be very useful for further complex and detailed calculations, such as solving boundary layer or Navier Stokes equations. It is not difficult to introduce different flow solvers for different domains or meshes, which could treat very complex problems both in geometry and flow properties. This is an important advantage of Chimera method.

Figures 14 and 15 show the Mach number contours of the solution of a flow over a sphere covered by four pieces of grid, at two different Mach numbers. One is at  $M_\infty = 0.2$  and the other is at its critical Mach number  $M_\infty = 0.55$ . Both figures show good agreement with potential solutions.

CONTOUR LEVELS

0.00000  
0.01000  
0.02000  
0.03000  
0.04000  
0.05000  
0.06000  
0.07000  
0.08000  
0.09000  
0.10000  
0.11000  
0.12000  
0.13000  
0.14000  
0.15000  
0.16000  
0.17000  
0.18000  
0.19000  
0.20000  
0.21000  
0.22000  
0.23000  
0.24000  
0.25000  
0.26000  
0.27000  
0.28000  
0.29000  
0.30000

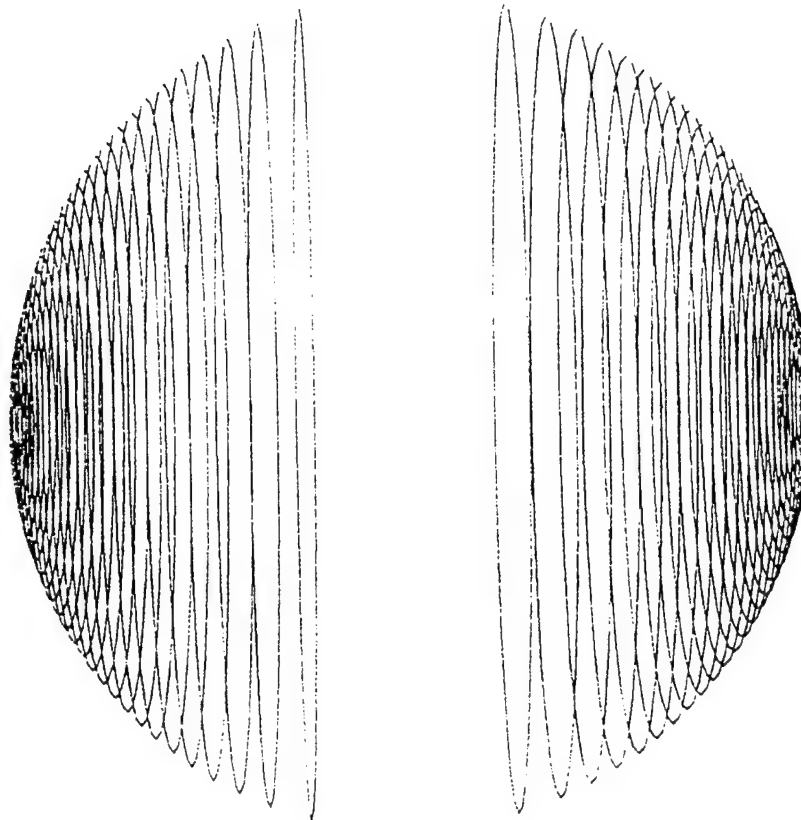


Figure 10. Potential Solution on a sphere Surface

CONTOUR LEVELS

0.00000  
0.01000  
0.02000  
0.03000  
0.04000  
0.05000  
0.06000  
0.07000  
0.08000  
0.09000  
0.10000  
0.11000  
0.12000  
0.13000  
0.14000  
0.15000  
0.16000  
0.17000  
0.18000  
0.19000  
0.20000  
0.21000  
0.22000  
0.23000  
0.24000  
0.25000  
0.26000  
0.27000  
0.28000  
0.29000  
0.30000

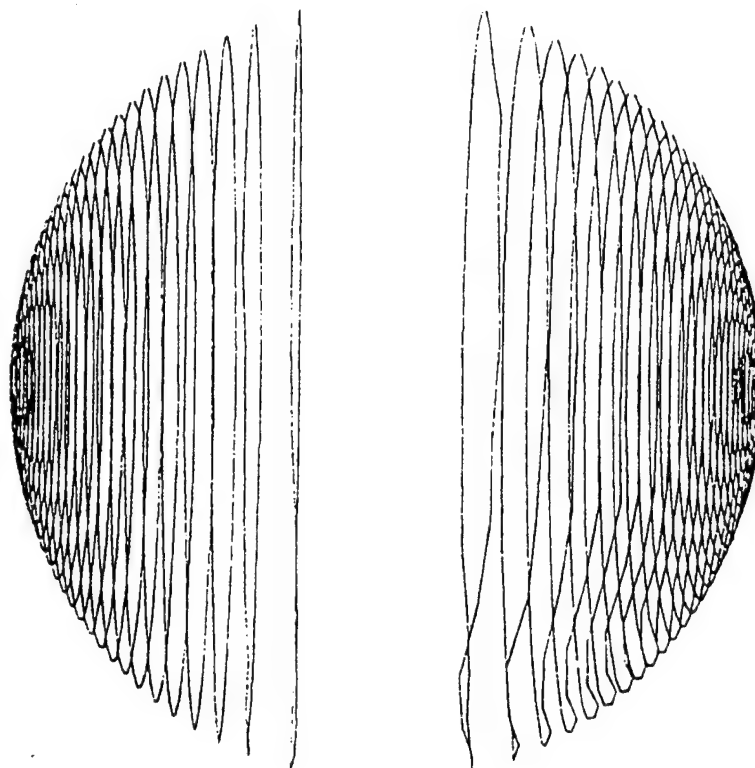


Figure 11. Numerical Solution on a sphere Surface.

CONTOUR LEVELS

) 00000  
 ) 01000  
 ) 02000  
 ) 03000  
 ) 04000  
 ) 05000  
 ) 06000  
 ) 07000  
 ) 08000  
 ) 09000  
 ) 10000  
 ) 11000  
 ) 12000  
 ) 13000  
 ) 14000  
 ) 15000  
 ) 16000  
 ) 17000  
 ) 18000  
 ) 19000  
 ) 20000  
 ) 21000  
 ) 22000  
 ) 23000  
 ) 24000  
 ) 25000  
 ) 26000  
 ) 27000  
 ) 28000  
 ) 29000  
 ) 30000

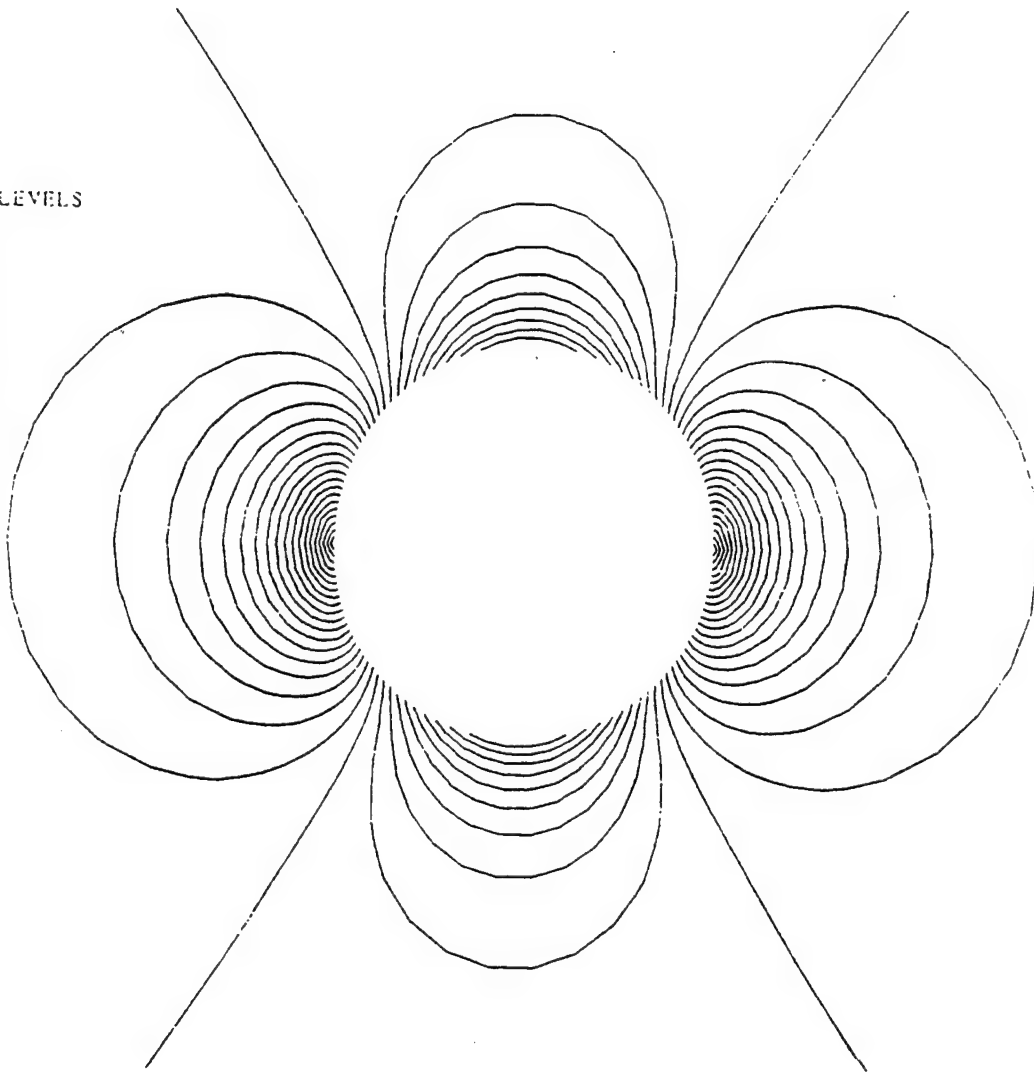


Figure 12. Potential Solution a sphere a Symmetric section

CONTOUR LEVELS

0.00000  
0.01000  
0.02000  
0.03000  
0.04000  
0.05000  
0.06000  
0.07000  
0.08000  
0.09000  
0.10000  
0.11000  
0.12000  
0.13000  
0.14000  
0.15000  
0.16000  
0.17000  
0.18000  
0.19000  
0.20000  
0.21000  
0.22000  
0.23000  
0.24000  
0.25000  
0.26000  
0.27000  
0.28000  
0.29000  
0.30000

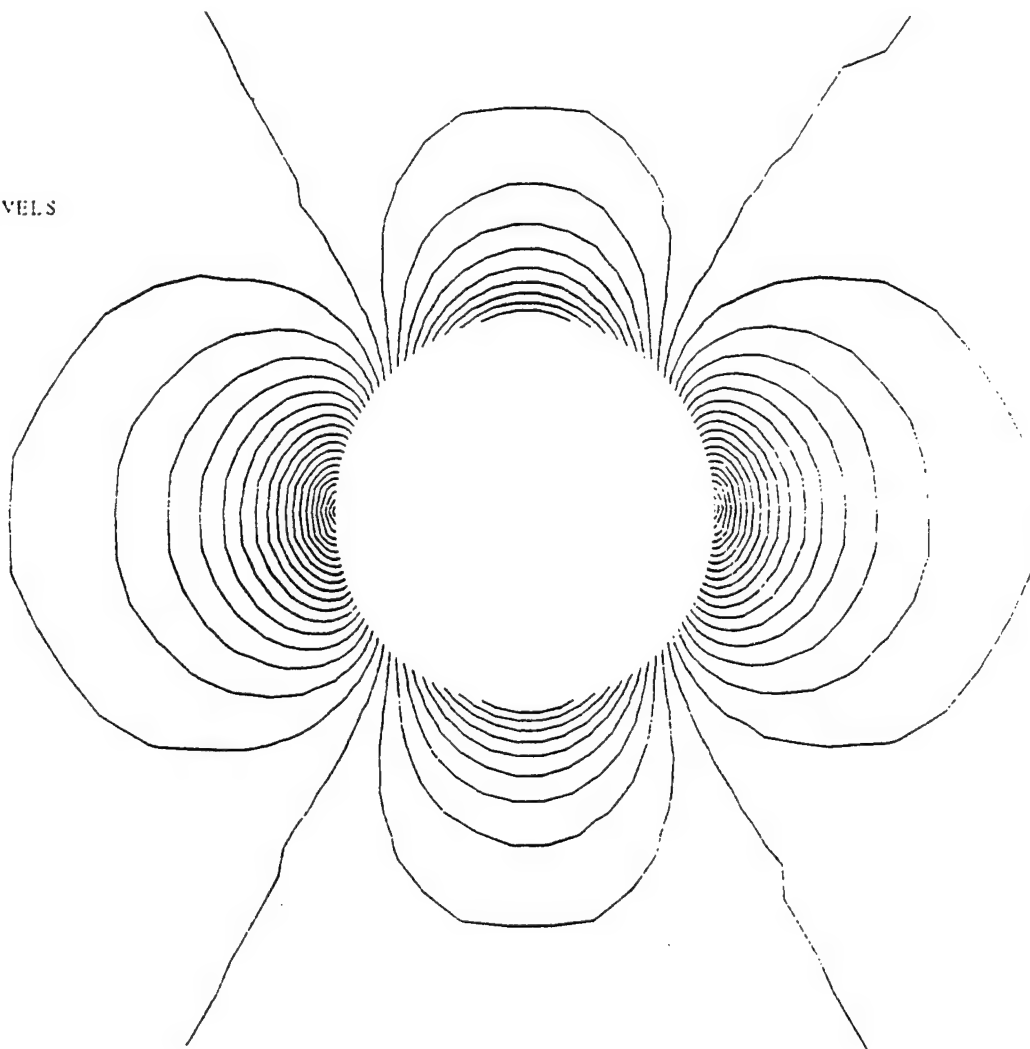


Figure 13. Numerical Solution on a sphere (a Symmetric section)

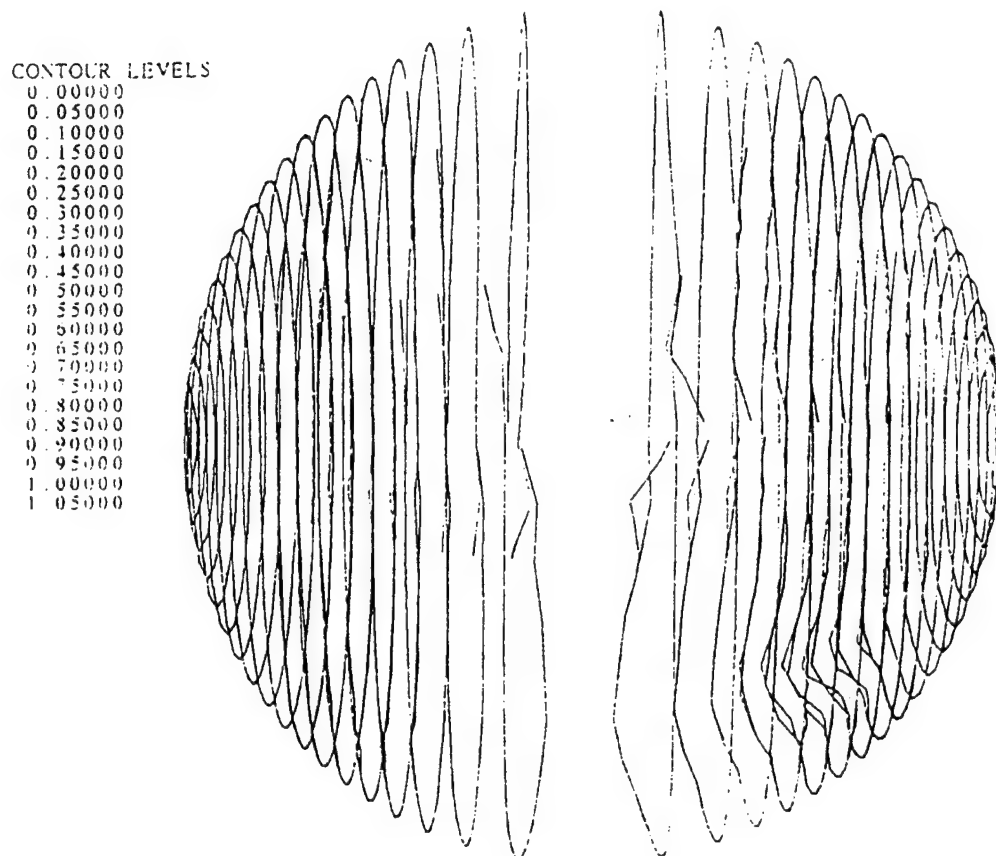


Figure 14. Mach Number Contour on Sphere Surface for four grids at  $M_{\infty} = 0.55$

CONTOUR LEVELS

0.01000  
0.02000  
0.03000  
0.04000  
0.05000  
0.06000  
0.07000  
0.08000  
0.09000  
0.10000  
0.11000  
0.12000  
0.12000  
0.12000  
0.14000  
0.15000  
0.16000  
0.17000  
0.18000  
0.19000  
0.20000  
0.21000  
0.22000  
0.22000  
0.22000  
0.24000  
0.25000  
0.26000  
0.27000  
0.28000  
0.29000  
0.30000  
0.31000

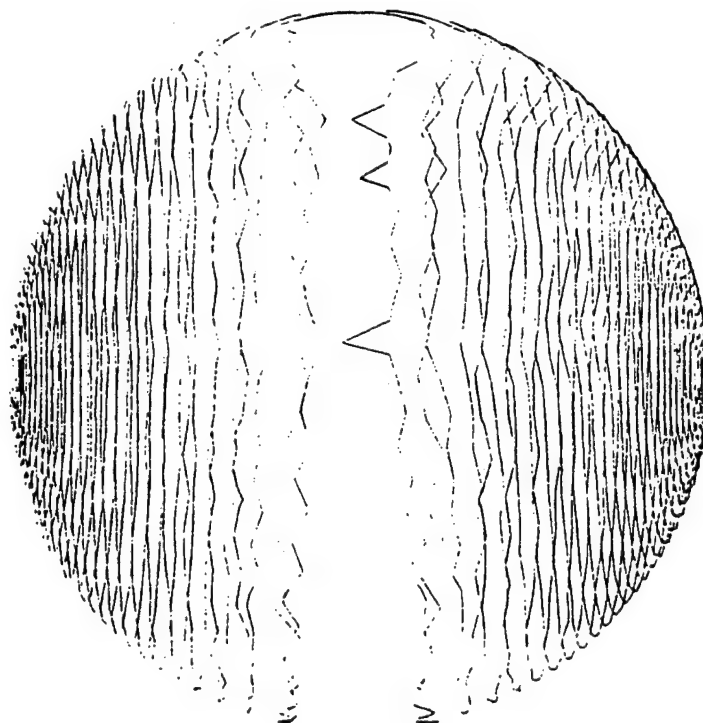


Figure 15. Mach Number Contour on Sphere Surface Interpolated from Four Grids

#### 4.4 Flow Solutions with Two-Body Configurations

In general, for complex multibody configurations problems, there will be some holes in the mesh when the solid bodies are embedded in the mesh. The simplest 3-D case is two spheres next to each other, and each grid is embedded in the other grid, as shown in a two-dimensional cross section in Figure 16. There is a hole in each mesh where the other sphere bodies are embedded. Figure 17 shows the velocity vectors in a symmetric section of the flow field for the two spheres. Two spheres and ellipsoids placed next to each other are also calculated as a test case at  $M_\infty = 0.20$ , which is in Figure 18 through 20. Both results are reasonable, therefore, it is possible to say that the code can handle multibody configuration problems.

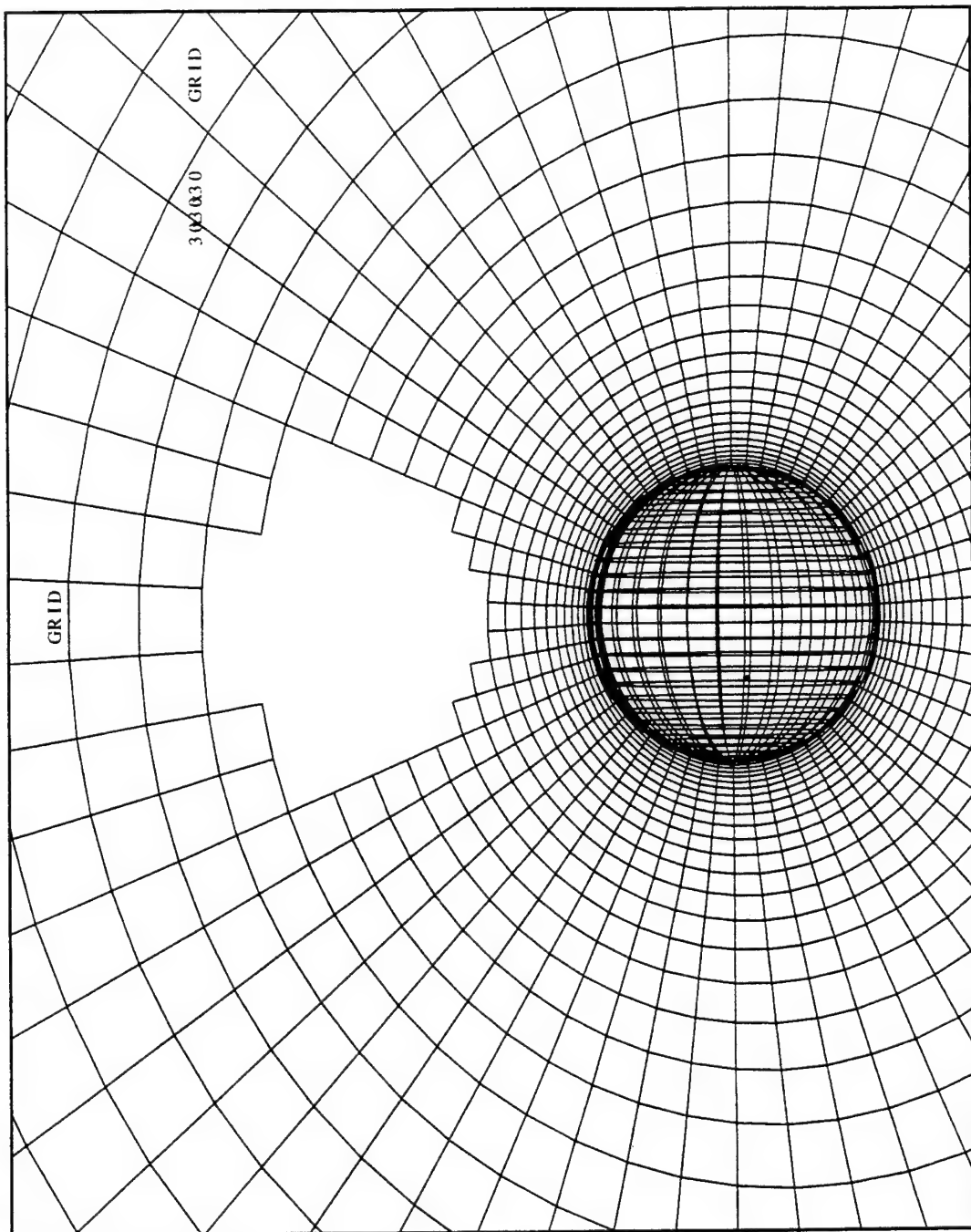


Figure 16: One of two grids with a hole in it for two sphere

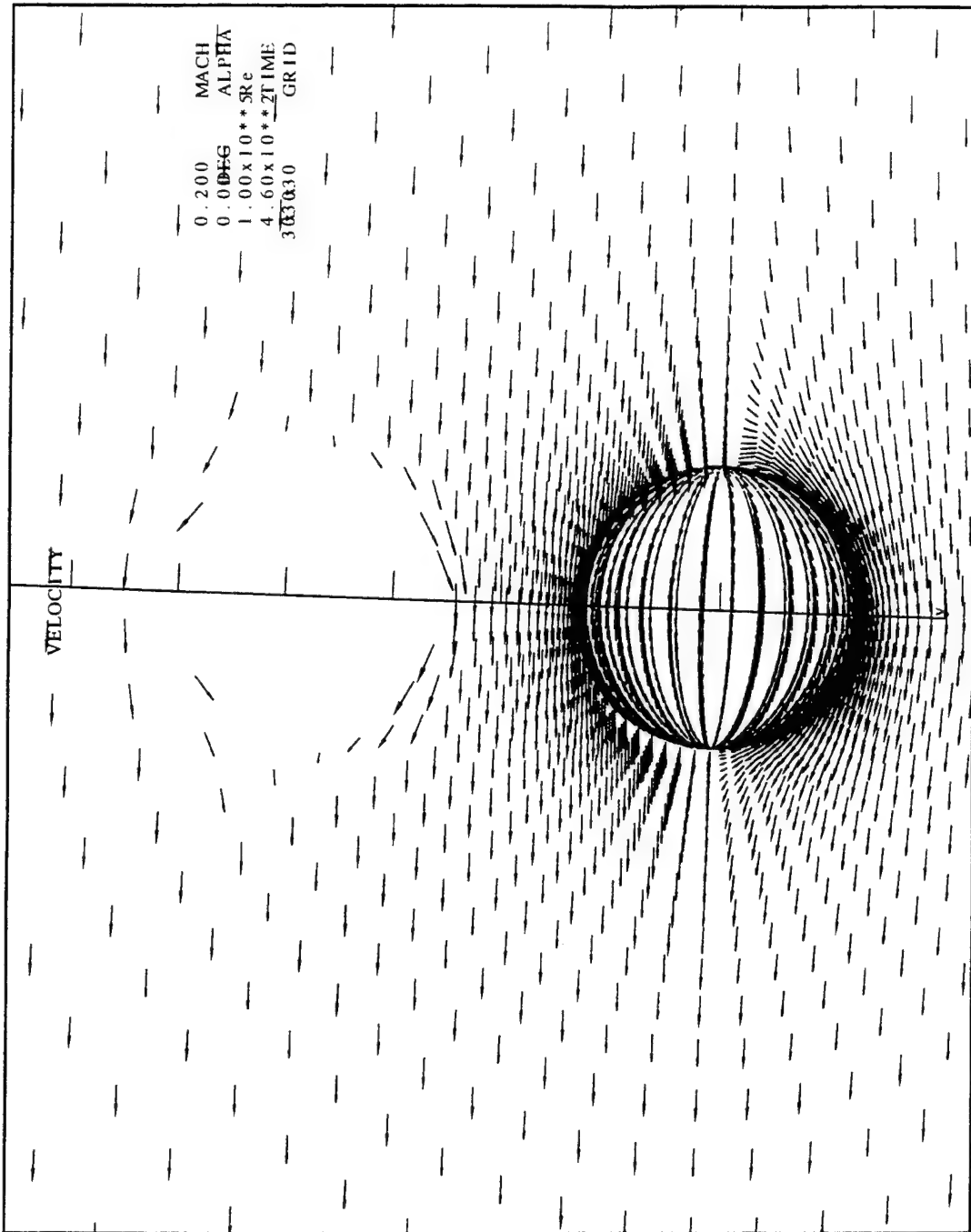


Figure 17: Velocity vector in a cross section of two spheres

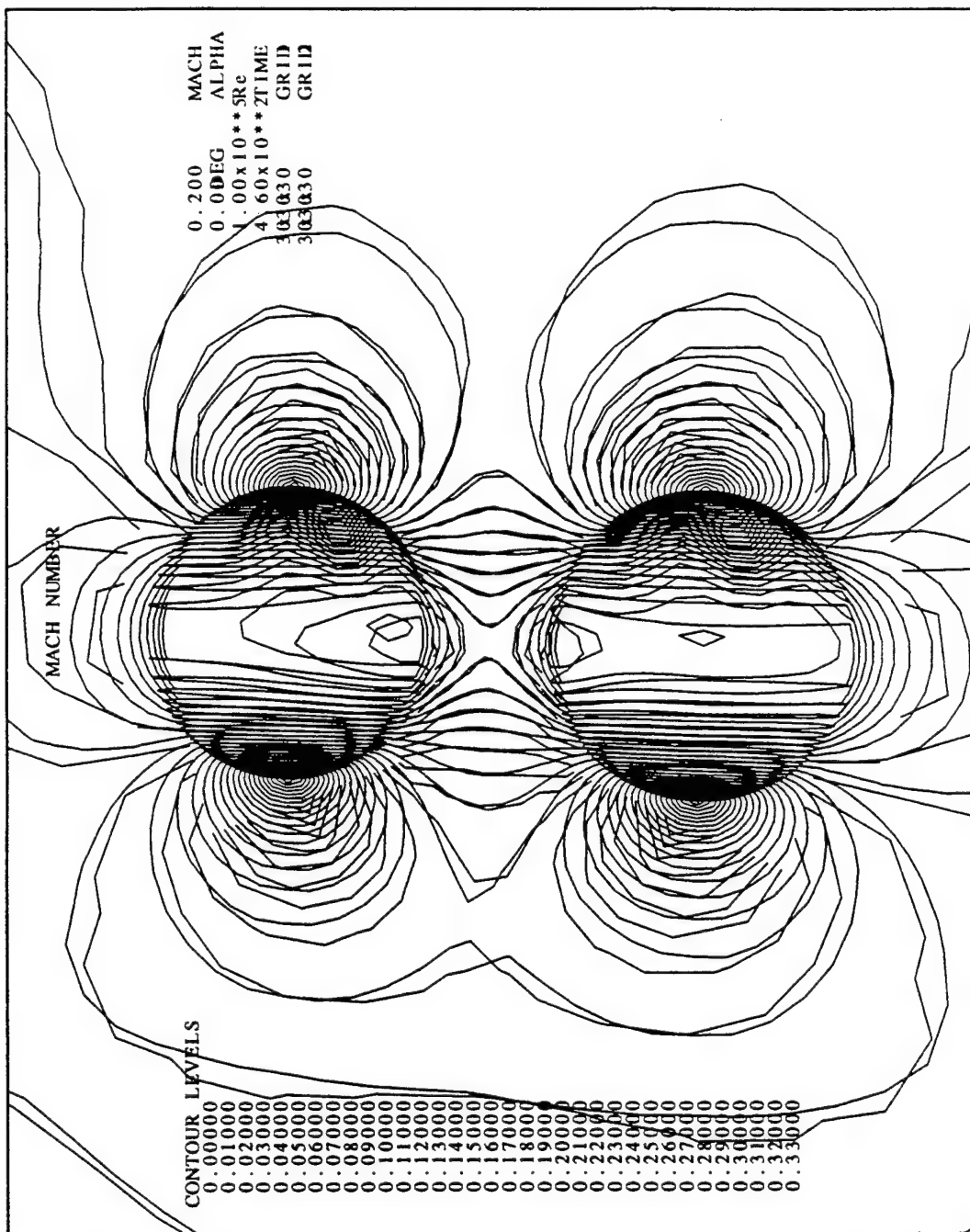


Figure 18: Pressure contour in a cross section of two spheres

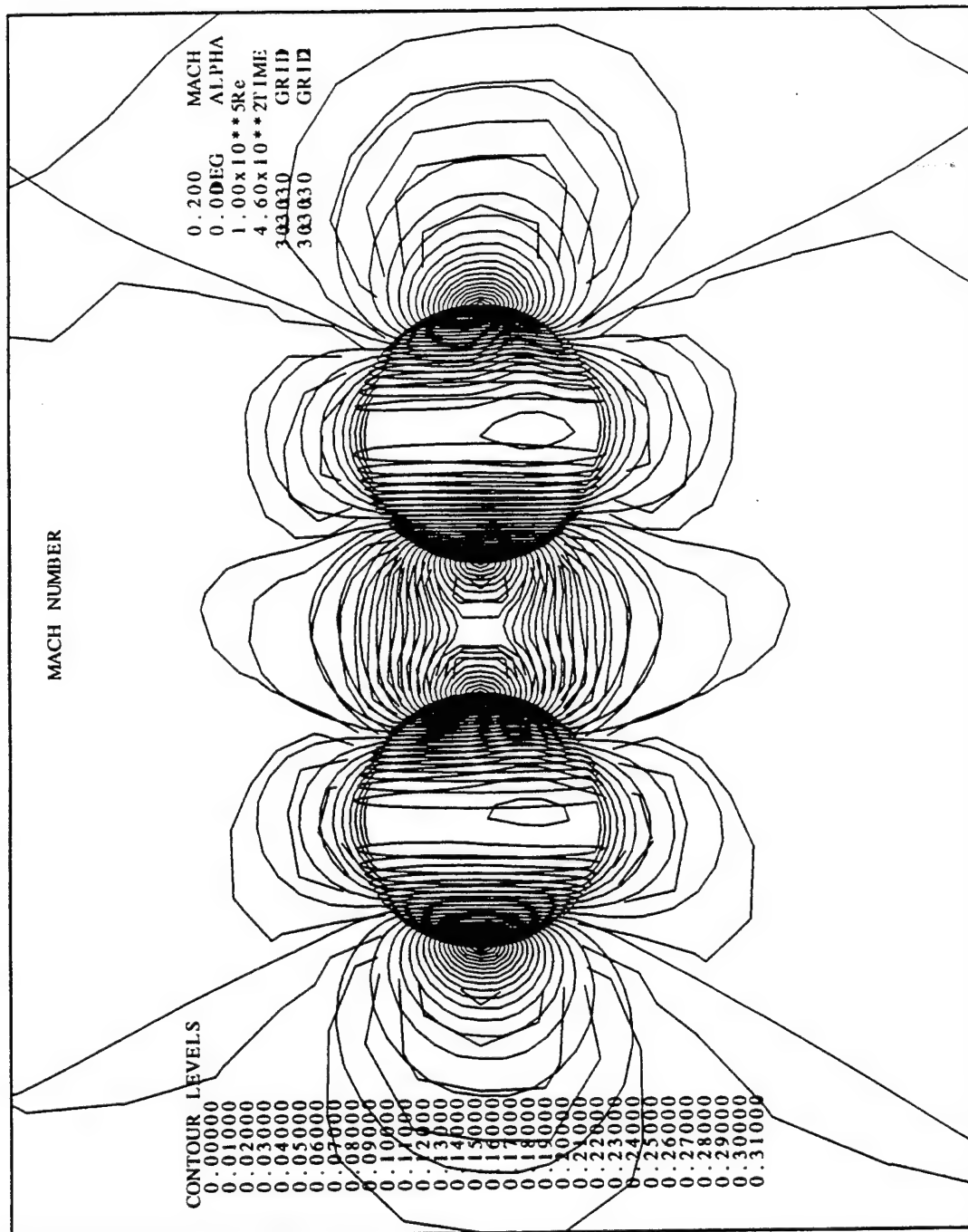


Figure 19: Pressure contour in a cross section of two spheres

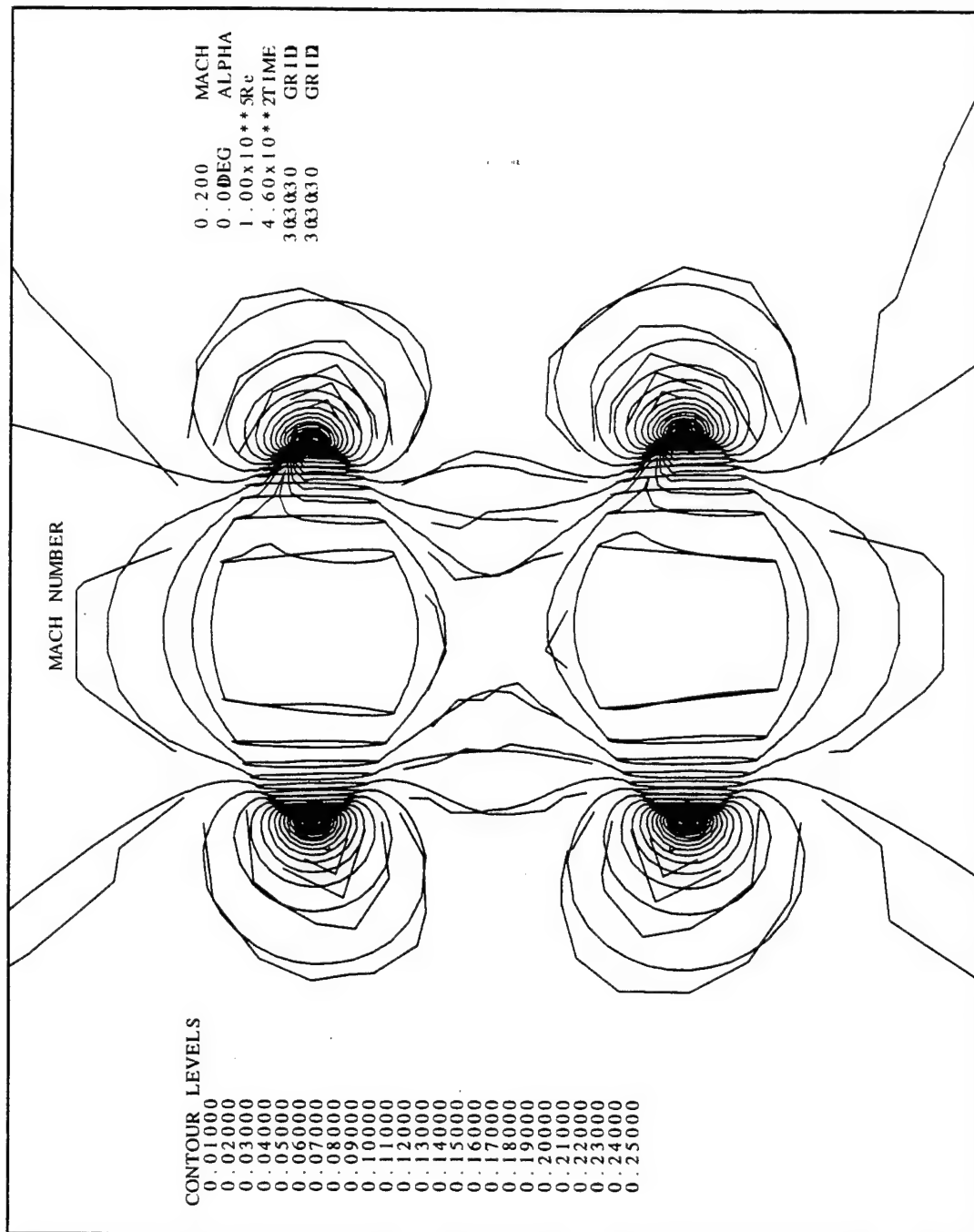


Figure 20: Pressure contour in a cross section of two ellipsoids

## 5.0 CONCLUDING REMARKS

A complete computer code has been described that implement 3-D grid-embedding techniques as a part of a flexible solution concept that we have called a Chimera method. The code utilize procedures for combining grids, locating embedded boundaries and interpolation points, and manipulating complex data structures. The validity of the method was successfully demonstrated on several geometries for inviscid flow.

A simple and robust algorithm for steady inviscid flow, in terms of primitive variables, is described for body fitted coordinates. This algorithm has been adapted for the Chimera framework for the calculation of complex geometric problems. The flow about a sphere and an ellipsoid, for both single and overset grids, showed good agreement with the exact solution for subcritical and critical cases. Comparison with an implicit Euler solver for the flow about an ellipsoid showed the computational advantages of the present method. Therefore, the flow solver presented here could possibly be quite useful for preliminary engineering analysis.

## 6.0 REFERENCES

1. Barth, T., "On Unstructured Grids and Solvers", von Karman Institute Lecture Series, Rhode-Saint-Genese, Belgium, March, 1990.
2. Benek, J. A., Buning, P. G. and Steger, J. L., "A 3-D Chimera Grid Embedding Technique", AIAA - 85-1523, Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, OH, July 15-17, 1985.
3. Benek, J., Donegan, T., and Suhs, N., "Extended Chimera Grid Embedding Scheme with Application to Viscous Scheme with Application to Viscous Flows", AIAA Paper 87-1126-CP, 1987
4. Benek, J. A., Steger, J. L., Dougherty, F. C., "A Flexible Grid Embedding Technique with Application to the Euler Equations", AIAA Paper No. 83-1944-CP, presented at the AIAA 6th Computational Fluid Conference, Danvers, MA, July, 1983
5. Benek, J. A., Steger, J. L., Dougherty, F.C. and Buning, P. G., "Chimera: A Grid Embedding Technique", AEDC-TR-85-64 Arnold Engineering Development Center, Arnold AF Station, TN, April 1986.
6. Chan, W.M. and Steger, J. L., "A Generalized Scheme for Three Dimensional Hyperbolic Grid Generation", AIAA 10th Computational Fluid Dynamics conference, June 24-26, 1991.
7. Chawla, K., Van Dalsem, W., and Rao, K., "Simulation and Analysis of a Delta Platform with Multiple Jets in Ground Effect", AIAA Paper 90-0299, January 1990.
8. Cordova, J Q. and Barth, T. J., "Grid Generation for General 2-D regions Using Hyperbolic Equations", AIAA Paper No. 88-0520, 1988.
9. Dietz, W., Jacocks, J., and For, J., "Application of Domain Decomposition to the Analysis of Complex Aerodynamics Configurations", SIAM Conf. Domain Decomposition Meths., Houston, TX, March, 1989.
10. Dougherty, F.C., Benek, J. A. and Steger, J. L., "On Applications of Chimera Grid Schemes to Store Separation", Proceedings of the AGARD Symposium on Store/Airframe Aerodynamics. Athens, Greece Oct. 8-10, 1985.
11. Dougherty, C. and Kuan, J., "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme", AIAA Paper 89-0637, January 1989.
12. Fox, J., Donegan, T., Jacocks, J., and Nichols, R., "Computation of the Euler Flow Field Produced about a Transonic Aircraft with Stores", AIAA Paper 89-2219. July-August 1989.

13. Fuji, J., "A Zonal Method for Practical Fluid Dynamics Problems", Fourth International Symposium on Computational Fluid Dynamics, University of California, Davis, Sept. 9-12, 1991.
14. Kinsey, D. W. and Barth, T. J., "Description of a Hyperbolic Grid Generation Procedure for Arbitrary Two-Dimensional Bodies", AFWAL TM 84-191-FIMM, 1984.
15. Kiris C. and Chang, I., "Numerical Simulation of the Incompressible Internal Flow Through a Tilting Disk Valve", AIAA Paper 90-0682, January 1990.
16. Meakin, R. L., "overset Grid Methods for Aerodynamic Simulation of Bodies in Relative Motion", 8th Aircraft/Store Compatibility Symposium, Fort Walton Beach, Florida, 1990.
17. Meakin, R. L., "Transient Flow Field Responses about the Space Shuttle Vehicle During Ascent and SRB Separation", R. Aero. Soc. Store Carriage, Integration, and Release Conf., pp 29.1-29.16, 1990.
18. Meakin, R. L., and Street, R., "Simulation of Environmental Flow Problems in Geometrically Complex Domains. Part 2: A Domain-Splitting Method", Comp. Meths. Appl. Mech. Engrg., 68, 311-331, 1988.
19. Meakin, R. and Suhs, N., "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion", AIAA Paper 89-1996, June 1989.
20. Moon, Y. J. and Liou, M.S., "Conservative Treatment of Boundary Interfaces for Overlain Grids and Multi-level Grid Adaptations", AIAA Paper 89-1980-CP, 1989.
21. NASA Ames Space Shuttle Flow Simulation Group. Buning, P.G., chiu, I. T., Martin, F.W., Meakin, R.L., Obayashi, S., Risk, Y.M., and Steger, J.L., "Flowfield Simulation of the Space Shuttle Vehicle in Ascent", Proceeding of the Fourth International Conference on Supercomputing, Santa Clara Convention Center, Santa Clara, CA April 30-May 5, 1989.
22. NASA Ames Space Shuttle Flow Simulation group, Buning P., Chiu, I., Mertin, F., Meakin R., Obayashi, S., Risk, Y., Steger, J., and Yarrow, M. "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent", 4th International Conference on Super Computing, Santa Clara, California, April, 1989.
23. NASA Ames Space Shuttle Flow Simulation Group, Buning, P.G., Chiu, I.T., Obayashi, S., Risk, Y.M., and Steger, J.L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent", AIAA 88-4359-CP, 1988.
24. Parks, S.J., Buning, P.G., Steger, J.L., and Chan, W.M., "Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme", AIAA-91-1587, June 24-26, 1991/Honolulu, HI
25. Risk, Y. and Schiff, L., "Numerical Simulation of the Viscous Flow Around a Simplified F/A-18 at High Angles of Attack", AIAA Paper 90-2999, August, 1990.
26. Rizk, Y.M., Schiff, L.B., and Gee, K. "Numerical Simulation of the Viscous Flow Around a Simplified f/a-18 at High Angle of Attack", AIAA Paper No. 90-2999, 1990.

27. Rizk, Y.M. and Steger, J.L. and Chaussee, D., "Use of a Hyperbolic Grid Generation Scheme in Simulating Supersonic Viscous Flow About Three-Dimensional Winged-Configurations", NASA TM 6776, 1985.
28. Starius, G., "Constructing Orthogonal Curvilinear Meshes by Solving Initial Value Problem", *Numerische Mathematik*, 28, 1977, pp. 25-48.
29. Steger, J.L., "Generation of Three-Dimensional Body-Fitted Grids by Solving Hyperbolic", *Partial Differential Equations*, NASA TM 101069, 1989.
30. Steger, J. L., "Notes on Surface Grid Generation Using Hyperbolic Partial Differential Equation", Technical Memorandum CFD/UCD 89-101.
31. Steger, J. L., and Benek, J. A., "On the Use of Composite Grid Schemes in Computational Aerodynamics", *Computer Methods in Applied Mechanics and Engineering* Vol. 64, Nos. 1-3, Oct. 1987.
32. Steger, J. L. and Chaussee, D. S., "Generation of Body-fitted Coordinates Using Hyperbolic Partial Differential Equations", *SIAM J. Sci. Stat. Comput.* 1, 1980. pp. 431-437.
33. Steger, J. L., Dougherty, F.C. and Benek, J. A., "A Chimera Grid Scheme, in *Advance in Grid Generation*", ASME, FED-Vol. 5, 1983.
34. Steger, J. L. and Rizk, Y.M., "Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations", NASA TM 86753, 1985.
35. Ronald L. Panton, "Incompressible Flow", John Willey and Sons, Inc., 1984.
36. Benek, J. A., Steger, J. L., and Dougherty, F. C., "A Flexible Grid Embedding Technique with Application to the Euler Equations", AIAA 83-1944, July, 1983.
37. Suhs, N. E., "Computations of Three-Dimensional Cavity Flow at Subsonic and Supersonic Mach Numbers", AIAA 87-1208, June 1987.
38. Dougherty, Carroll F., "Development of A Chimera Grid Scheme with Applications to Unsteady Problems", Ph.D. Dissertation of Stanford University, June, 1985.
39. Kutler, P., Steger, J.L. and Bailey, F. R., "Status of Computational Fluid Dynamics in the United States", AIAA-87-1135-CP, June, 1987.
40. Nakahashi, K., "FDM-FEM Zonal Approach for Computations of Compressible Viscous Flows", 10th International Conference on Numerical Methods in Fluid Dynamics. Beijing, China, June 1986.
41. Nakahashi, K. and Obayashi, S., "Viscous Flow Computations Using a Composite Grid", AIAA Paper No. 87-1128-CP, AIAA 8th Computational Fluid Dynamics conference, 1987.
42. Weatherhill, N. P., "On the Combination of Structured-Unstructured Meshes", in *Numerical Grid Generation in Computational Fluid Dynamics*, S. Sengupta, ed., Pineridge Press (1988).

43. Lasinski, T. A. et al., "Computation of the Steady Viscous Flow Over a Tri-Element Augmentor Wing Airfoil", AIAA Paper No. 82-0021, January 1982.
44. Hennesius, K. A. and Pulliam, T. H., "A Zonal Approach to Solutions of the Euler Equations", AIAA Paper. 82-0969, June 1982.
45. Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations", AIAA Paper. 84-0164, January 1984.
46. Miki, K. and Takagi, T., "A Domain Decomposition and Overlapping Method for the Generation of Three-Dimensional Boundary-Fitted Coordinate Systems", Journal of Computational Physics, Vol. 53, No. 2, February 1982, pp. 319-330.
47. Holst, T.L. et al, "Numerical Solution of Transonic Wing Flows Using an Euler/Navier-Stokes Zonal Approach", AIAA Paper. 85-1640, July 1985.
48. Gnoffo, P.A., "A Vectorized, Finite Volume, Adaptive Grid Algorithm for Navier-Stokes", Numerical Grid Generation, J. F. Thompson, ed., North-Holland, New York, New York, 1982, pp. 819-836.
49. Nakahashi, K. and Deivert, G. S., "A Self-Adaptive-Grid Method with Application to Airfoil Flow", AIAA Paper. 85-1525, July 1985.
50. Ghia, K., Ghia, U., and Shin, C. T., "Adaptive Grid Generation for Flows with Local High Gradient Regions", Advances in Grid Generation, K. N. Ghia and U. Ghia, eds., ASME-FED-Vol. 5, June 1983.
51. Brackbill, J. V., "Coordinate System Control: Adaptive Meshes", Numerical Grid Generation, J. F. Thompson, ed. North-Holland, New York, New York, 1982, pp. 277-294.
52. Salzman, J. and Brackbill, J. V., "Application and Generation of Variational Methods for Generating Adaptive Systems", Numerical Grid Generation, J. F. Thompson, ed. North-Holland, New York, New York, 1982, pp. 865-884.
53. Berger, J. J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations", Stanford University STAN-CS-82-924, August 1982.
54. Berger, J. J. and Oliger, J., "Adaptive Mesh REfinement for Hyperbolic Partial Differential Equations", Journal of Computational Physics, Vol. 53, No. 3, March 1984, pp. 484-512.
55. Steger, J. L., "CLass Textbook for Computational Fluid Dynamics", University of California, Davis, Spring, 1990.
56. Flores, J., Reznick, S. G., Holst, T. L., and Gundy, K. L., "Transonic Navier-Stokes Solution for a Fighter-like Configuration", AIAA Paper 87-0032.
57. Obayashi, S., "Toward the Navier-Stokes Analysis of Transport Aircraft Configurations", AIAA Paper 87-0428.
58. Jameson, A. and Baker, T. J., "Improvements to the Aircraft Euler MEthod", AIAA Paper 87-0452.